

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Fakulta kybernetiky a biomedicínského inženýrství

Zjištění pohybu předmětu v prostoru více-kamerovým systémem
Detection of the Movement of the Object in the Space with a Multi-camera System

Zadání bakalářské práce

Student: **Simona Dárková**
Studijní program: B2649 Elektrotechnika
Studijní obor: 2612R041 Řídicí a informační systémy
Téma: Zjištění pohybu předmětu v prostoru více-kamerovým systémem
Detection of the Movement of the Object in the Space
with a Multi-camera System

Jazyk vypracování: čeština

Zásady pro vypracování:

Cílem práce je seznámení se se základními programovacími technikami a matematickými modely zpracování obrazu a porozumění problematice snímání obrazu pomocí kamer. V počítači je takto nasnímaný obraz zpracováván pomocí softwaru pro strojové vidění. Výsledkem bude praktická realizace systému pro výpočet a vizualizaci pohybu předmětu v prostoru.

Body zadání:

1. Seznámení se s programovým prostředím LabVIEW, popis vývojových prostředí pro zpracování obrazu.
2. Rozbor funkce a popis algoritmů strojového zpracování obrazu za účelem sledování předmětu, segmentace obrazu a detekci objektů.
3. Návrh klíčových algoritmů a výsledné kamerové inspekce.
4. Návrh a vývoj SW komponent aplikace v programovacím prostředí LabVIEW.
5. Ověření funkce a zhodnocení.

Seznam doporučené odborné literatury:

- [1] VLACH, Jaroslav, Josef HAVLÍČEK a Martin VLACH. *Začínáme s LabVIEW*. Praha: BEN – technická literatura, 2008, 247 s. ISBN 978-80-7300-245-9.
- [2] HLAVÁČ, Václav a Milan ŠONKA. *Počítačové vidění*. Praha: Grada, 1992. ISBN 80-85424-67-3.
- [3] ŠONKA, Milan, Václav HLAVÁČ a Roger BOYLE. *Image processing, analysis, and machine vision*. 3rd ed. Toronto: Thomson, 2008, xxv, 829 s. ISBN 978-0-495-08252-1. Dostupné také z: <http://www.engineering.uiowa.edu/~dip/LECTURE/lecture.html>.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Pavel Kodytek**

Datum zadání: 01.09.2018

Datum odevzdání: 30.04.2019



doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry

prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovávala samostatně. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

V Ostravě dne: 29. dubna 2019

A handwritten signature in black ink, appearing to be 'P. H.', written over a horizontal dotted line.

podpis studenta

Poděkování

Mé poděkování patří panu **Ing. Pavlu Kodytkovi** za odborné vedení, ochotu a především trpělivost, kterou mi v průběhu zpracování mé bakalářské práce věnoval. Dále bych ráda poděkovala panu **doc. Ing. Petru Bilíkovi, Ph.D.** za to, že dohlížel na tvorbu mé práce v případech, kdy byl Ing. Kodytek pracovně pryč.

Abstrakt

Tato bakalářská práce se zabývá snímáním předmětu v prostoru pomocí více-kamerového systému. Snímaný obraz je v reálném čase přenášén do vytvořeného softwaru pro strojové vidění, kde následně dochází k vybraným operacím pro úpravu obrazu. Pohyb detekovaného objektu je zobrazován do 3D grafu. Motivací k vytvoření právě této práce mi bylo tzv. jestřábí oko. Je to soustava několika speciálně kalibrovaných kamer, jejichž záznam je zpracováván speciálně vyvinutým softwarem. Tato technologie je v současnosti velkým pomocníkem v mnoha sportovních odvětvích.

Klíčová slova

LabVIEW, zpracování obrazu, kamerové systémy

Abstract

This thesis is about Detection of the Movement of the Object in the Space with a Multi-camera System. The captured image is transmitted in real time to the created machine vision software, where selected image editing operation are subsequently performed. The motion of the detected objects is displayed in a 3D graph. The motivation for creating this work was so – called Hawk – Eye. It is a set of several specially calibrated cameras whose recording is processed by specially developer software. This technology is currently a great help in many sports sectors.

Key words

LabVIEW, image procesing, multi-camera system

OBSAH

ÚVOD.....	13
1 Kamerové systémy.....	14
1.1 Rozdělení kamer podle způsobu přenosu signálu	15
1.1.1 Analogové kamery	15
1.1.2 Digitální IP kamery	15
1.2 Parametry kamer	15
1.2.1 Rozlišení.....	16
1.2.2 Typ snímače	16
1.2.3 Světelná citlivost	19
1.3 Objektiv.....	20
1.4 Komunikační rozhraní.....	20
1.4.1 USB 2.0 a USB 3.0.....	22
1.4.2 IEEE 1394 a/b	22
1.4.3 GigE Vision.....	22
1.4.4 Camera Link.....	22
2 Vývojové prostředí LabVIEW	23
2.1 Historie a vývoj	23
2.2 Princip a využití LabVIEW	24
3 Základní pojmy a operace s obrazem	25
3.1 Jasová reprezentace obrazu	25
3.1.1 Model RGB	26
3.1.2 Model CMYK	27
3.1.3 Model HSV	28
3.2 Jasové transformace obrazu.....	29
3.2.1 Globální jasová transformace	30
3.2.2 Lokální jasová transformace.....	30
3.2.3 Bodová jasová transformace.....	30
3.3 Geometrická transformace obrazu.....	32
3.3.1 Rotace.....	33
3.3.2 Změna měřítka.....	34
3.3.3 Posunutí.....	34

3.3.4 Zkosení	34
3.3.5 Složené transformace.....	35
3.4 Matematické morfologie	35
3.4.1 Dilatace	37
3.4.2 Eroze	37
3.5 Segmentace obrazu.....	38
3.5.1 Prahování.....	38
3.5.2 Detekce hran.....	39
4 Připojení kamery do LabVIEW	42
4.1 Webová kamera LifeCam Studio Q2F-00018	42
4.2 Komunikace webové kamery a LabVIEW	43
5 Pracovní prostředí.....	44
5. 1 Snímaný prostor	44
5. 2 Detekovaný objekt.....	46
6 Aplikace v LabVIEW.....	47
6. 1 Rozbor jednotlivých částí	47
6. 1. 1 Načtení kamery.....	48
6. 1. 2 Sejmутí referenčního pozadí	48
6. 1. 3 Porovnání	49
6. 1. 4 Vyznačení hledaného objektu.....	50
6. 1. 5 Vykreslení do grafu	51
6. 1. 6 Nenačtení kamery a ukončení.....	51
6. 1. 7 Ukončení programu	52
6. 2 Kód pro jednu kameru.....	52
6. 3 Funkčnost programu.....	53
6.4 Čelní panel aplikace	55
7 Práce s obrazem.....	56
7. 1 Výsledný obraz.....	56
7. 2 Vykreslení do grafu	59
ZÁVĚR	60
POUŽITÁ LITERATURA	61
Seznam příloh.....	65

Seznam použitých zkratk

VI – virtual instrument – základní jednotka aplikace v LabVIEW tvořena čelním panelem a blokovým diagramem

CMOS – complementary metal–oxide–semiconductor – typ snímače

PoCL – power of camera link – napájení přímo z komunikačního rozhraní

Seznam obrázků

Obrázek 1: Komerový systém [39]	14
Obrázek 2: Rozlišení kamer [39]	16
Obrázek 3: Princip činnosti CCD snímače [3]	18
Obrázek 4: Architektura CCD snímače [3]	18
Obrázek 5: Architektura CMOS snímače [3]	19
Obrázek 6: Vývoj grafického prostředí LabVIEW [1]	23
Obrázek 7: LabVIEW 2018	24
Obrázek 8: Chromatický diagram [11]	25
Obrázek 9 Model RGB [9]	27
Obrázek 10: Model CMY [15]	28
Obrázek 11: Model HSV a vyznačení jednotlivých složek a základních barev [15]	29
Obrázek 12: Znázornění jasové transformace [18]	29
Obrázek 13: Typy operací podle vlivu vstupního obrazu na výstupní obraz [19]	29
Obrázek 14: Převodní charakteristiky [18]	31
Obrázek 15: Ukázka histogramu vytvořeného v LabVIEW	31
Obrázek 16: Relativní srovnání jasu pro lidské oko [21]	32
Obrázek 17: Grafické znázornění jednotlivých geometrických transformací - změna měřítka, zkosení, rotace a posunutí [25]	33
Obrázek 18: Příklad bodové množiny $U = \{(1,0), (1,1), (0,2), (1,2), (2,2), (2,3), (3,3), (2,4)\}$ [4]	36
Obrázek 19: Příklady strukturních elementů [4]	36
Obrázek 20: Příklad dilatace (vlevo původní množina U , uprostřed množina B , vpravo výsledná množina po dilataci $U \oplus B$) [4]	37
Obrázek 21: Příklad eroze (vlevo původní množina U , uprostřed množina B , vpravo výsledná množina po erozi $U \ominus B$) [4]	38
Obrázek 22: Příklad prahování	39
Obrázek 23: Grafická definice gradientu [26]	40
Obrázek 24: Znázornění některých hran: vlevo skoková hrana, považována za ideální; vpravo zašuměna hrana, již lze nalézt v reálném obraze [26]	40
Obrázek 25: Vlevo původní funkce, uprostřed první derivace, vpravo druhá derivace [26]	41
Obrázek 26: Microsoft LifeCam Studio Q2F - 00018 [8]	42
Obrázek 27: NI MAX	43
Obrázek 28: Parametry konstrukce	44
Obrázek 29: Rozmístění kamer na konstrukci	45
Obrázek 30: Reálný prostor	45
Obrázek 31: Barevný obrázek uložen z LabVIEW	46
Obrázek 32: Barevný obrázek pořízen mobilním telefonem	46
Obrázek 33: Nastavení reentrantnosti	47
Obrázek 34: SubVI pro načtení kamery	48
Obrázek 35: SubVI pro sejmutí referenčního pozadí	48
Obrázek 36: Rozdíl aktuálního snímku a uloženého snímku a následná segmentace	49
Obrázek 37: Převod obrázků na pole	49

Obrázek 38: Práhování	50
Obrázek 39: SubVI pro vyznačení nejjasnějšího objektu	50
Obrázek 40: Vykreslení posledních 5hodnot do 3D grafu	51
Obrázek 41: SubVI vypisující nenačtení kamery a konec programu	51
Obrázek 42: Event struktura pro ukončení programu	52
Obrázek 43: Kód pro jednu kameru	52
Obrázek 44: Vývojový diagram mé aplikace v LabVIEW	54
Obrázek 45: Uživatelské rozhraní	55
Obrázek 46: Snímek s nízkou prahovací hodnotou	56
Obrázek 47: Snímek s vysokou prahovací hodnotou	56
Obrázek 48: Snímek z čelní kamery před prahováním - nárys	57
Obrázek 49: Snímek z čelní kamery po prahování - nárys	57
Obrázek 50: Snímek z boční kamery před prahováním - bokorys	58
Obrázek 51: Snímek z boční kamery po prahování - bokorys	58
Obrázek 52: 3D graf.....	59

Seznam tabulek

Tabulka 1: Porovnání snímačů CCD a CMOS [3].....	17
Tabulka 2: Hodnoty osvětlení v různých prostředích [42].....	20
Tabulka 3: Vlastnosti jednotlivých druhů rozhraní [3]	21
Tabulka 4: Kódování vybraných barev ve formě hexadecimální a dekadického kódu [14].....	26
Tabulka 5: Základní pojmy teorie množin [4]	35

ÚVOD

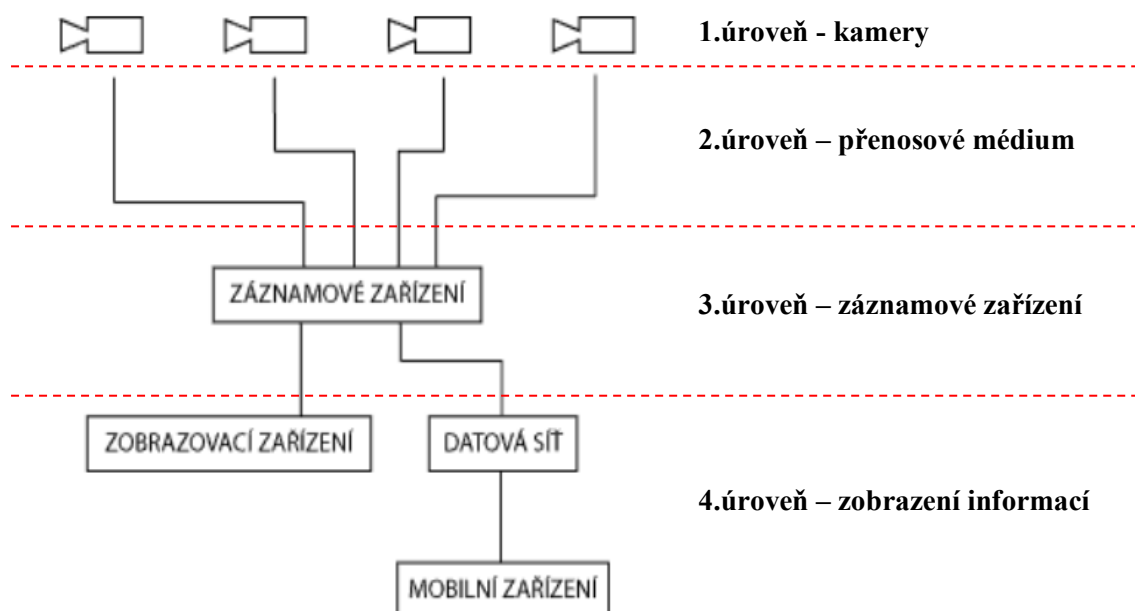
Cílem této práce je realizace a porozumění více kamerového systému zabývajícího se sledováním pohybujícího se objektu v prostoru. S rozvojem kamerových systémů v posledních letech, se tyto technologie začaly čím dál tím více využívat v sportovním odvětví, kde pomáhají lidem (například rozhodčím) určit pro člověka mnohdy obtížná rozhodnutí, která jsou založena na subjektivním vnímání. Pomáhají tedy kvantifikovat subjektivní rozhodnutí člověka, a zlepšují tedy kvalitu a spravedlnost sportovních výkonů. V moderní automatizaci a průmyslové praxi tyto systémy nachází také uplatnění, například v metodách high speed countingu, kde se více kamerový vysoko rychlostní systém dívá na místo průchodu částic a následně zaznamenává každou jednu z nich

Teoretická část mé bakalářské práce je rozdělena do 3 základních kapitol. První z nich se zabývá kamerovými systémy, jejich použitím, několika základním rozdělením především na analogové kamerové systémy a digitální IP kamerové systémy. Dále jsou zmíněny některé parametry výrazně ovlivňující kvalitu obrazu. Zbytek kapitoly je věnován nejčastěji používaným komunikačním rozhraním. Ve druhé kapitole je krátce popsáno vývojové prostředí LabVIEW. Třetí kapitola se zabývá základními pojmy používaným při zpracování obrazu. Je zde zmínka o nejčastěji využívaných barevných modelech nebo o úpravách prováděných pro dosažení co nejkvalitnějšího obrazu.

Praktická část je složena ze 4 kapitol. Nejprve je popsána kamera, kterou jsem při realizaci své práce využívala. V této kapitole je také zmíněno připojení kamery k notebooku/počítači a co je třeba uskutečnit pro tvorbu aplikace v LabVIEW. V další kapitole je blíže popsáno pracovní prostředí, tedy konstrukce na níž byly kamery přichyceny, jaké pozadí bylo zvoleno a proč, a také jaký objekt byl vybrán k detekci. Poté následuje kapitola, kde je detailně rozebrána aplikace vytvořená v LabVIEW. V této kapitole jsou popsána všechna subVI. Dále kapitola nabízí pohled na uživatelské prostředí aplikace. Poslední kapitola je věnována konkrétním úpravám a operacím, jež byly s obrazem prováděny. Jsou zde zobrazeny výsledky, kterých bylo dosaženo, a to jak ve formě jednotlivých snímků, tak ve formě 3D grafu.

1 Komerové systémy

První kapitola je věnována základnímu popisu kamerových systému. Tento systém se dá rozdělit do 4 základních úrovní, viz Obrázek 1. Na nejvyšší úrovni jsou samotné kamery. Na druhé úrovni je pak přenosové médium, jímž jsou přenášeny informace jdoucí z kamery. Na úrovni třetí se nachází záznamové zařízení, které se stará o zpracování signálu. Nakonec je tento signál předáván buďto přímo do zobrazovacího zařízení nebo do lokální datové sítě (LAN), odkud může být zobrazen jakýmkoliv počítačem nebo mobilním zařízením v této síti.



Obrázek 1: Kamerový systém [39]

V současnosti je kamerových systému využíváno velmi často, a to nejčastěji jako zabezpečovacího prvku, jak už pro soukromé účely, tak například k monitorování veřejných objektů. Dalším využitím může být například sport.

Na rozdělení kamerových systémů může být nahlíženo z různých úhlů, ale jelikož to není stěžejním tématem této práce, uvedu zde pouze základní rozdělení podle způsobu přenosu signálu. Jiná rozdělení by pak mohla být podle prostředí, kde jsou kamery využívány, podle světelných podmínek, do nichž jsou kamery určeny, podle způsobu snímání, tedy barevné rozlišení obrazu nebo podle konstrukčního provedení, to se odvíjí od prostředí, do něhož jsou instalovány.

Následující podkapitoly jsou zaměřeny na 1. a 2. úroveň kamerových systémů, . Nejprve jsou blíže rozebrány důležité parametry kamery a následně zmínka o nezbytné součásti samotné kamery, objektivu. Poslední část je věnována vybraným komunikačním rozhraním.

1.1 Rozdělení kamer podle způsobu přenosu signálu

Jedním z hlavním rozdělení kamer je podle způsobu, jakým přenáší snímání signál. Jsou to kamery analogové a digitální IP kamery.

1.1.1 Analogové kamery

CCTV (Close Circuit TeleVision) neboli analogové kamery jsou stále ještě nejpoužívanějším typem při snímání obrazu. Princip analogových kamer je takový, že informace o zvuku i obrazu jsou na pásku zaznamenávány různou intenzitou zmagnetizování povrchu pásky. Při zpětné rekonstrukci je pak měřena intenzita zmagnetizování povrchu pásky. Tento způsob záznamu a čtení informací nikdy nebude 100% a při každé další kopii bude záznam více znehodnocen šumem.

Rozlišení těchto kamer bývá do 0,4 MPix a snímkovácí frekvence maximálně 25 fps. Kamery mohou být jak černobílé, tak i barevné. Signál z těchto kamer je přenášen přes koaxiální kabel do záznamového zařízení. Napájení kamer je řešeno zvlášť. Ve více-kamerovém systému mohou mít všechny kamery společný zdroj, nebo každá kamera může být vybavena vlastním zdrojem.

[39],[40],[41]

1.1.2 Digitální IP kamery

Digitální IP kamery jsou nastupující variantou v kamerových systémech. Jak název napovídá, tyto kamery používají pro komunikaci se záznamovým zařízením počítačovou IP síť. Každá kamera má svou vlastní IP adresu, při splnění určitých podmínek lze tedy sledovat obraz z jakékoliv kamery. Signál z těchto kamer je převáděn do digitální podoby, která je vyjádřena čísly a ve formě jedniček a nul je zapisována na pásku. Tento způsob zápisu je daleko výhodnější, jelikož téměř nemůže dojít k chybnému přečtení informace z pásky.

Kamery tohoto typu se vyrábějí většinou se snímačem typu CMOS, na rozdíl od kamer analogových, které bývají vybaveny snímačem CCD. Nevýhodou digitálních kamer je menší citlivost, to je ovšem vynahrazováno větším rozlišením, které se pohybuje standardně od 1,3 do 2 MPix. Snímkovácí frekvence kamer je v rozmezí 6 – 60 fps. Napájení kamery je většinou řešeno přes Ethernet (PoE – Power of Ethernet).

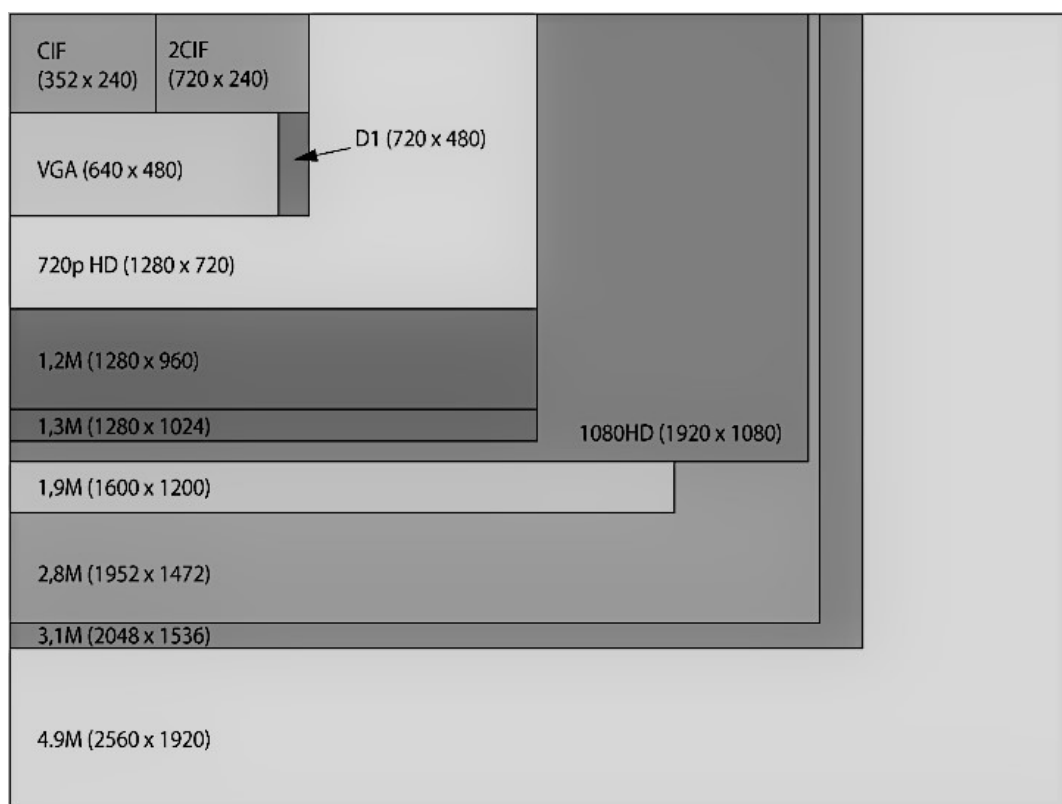
[39],[40],[41]

1.2 Parametry kamer

Parametrů, jež by mohly být zmíněny, je nespočet. Já jsem vybrala rozlišení kamer, které říká, v jaké kvalitě bude pořízený obraz či záznam zobrazen nebo uložen. Druhým parametrem je typ snímacího prvku, jímž jsou kamery vybaveny. Posledním parametrem je světelná citlivost, rovněž vysoce ovlivňující kvalitu obrazu.

1.2.1 Rozlišení

Rozlišení kamer vychází z rozlišovací schopnosti snímače v kameře. V případě analogu je obraz složen z tzv. TV řádků, jelikož tento typ vychází z televizního průmyslu. V případě digitálu je obraz složen z pixelů (čtvercové obrazové body). První číslo je počet pixelů v řádku, druhé číslo je počet pixelů ve sloupci. Výsledkem po vynásobení těchto dvou hodnot je pak celkový počet pixelů v obrazu.



Obrázek 2: Rozlišení kamer [39]

1.2.2 Typ snímače

Každá kamera je vybavena určitým typem snímače. V současnosti se rozlišují dva základní typy snímačů, CCD a CMOS. Oba tyto snímače jsou založeny na principu fotoelektrického jevu. V Tabulka 1 stručné porovnání těchto snímačů. [3]

Tabulka 1: Porovnání snímačů CCD a CMOS [3]

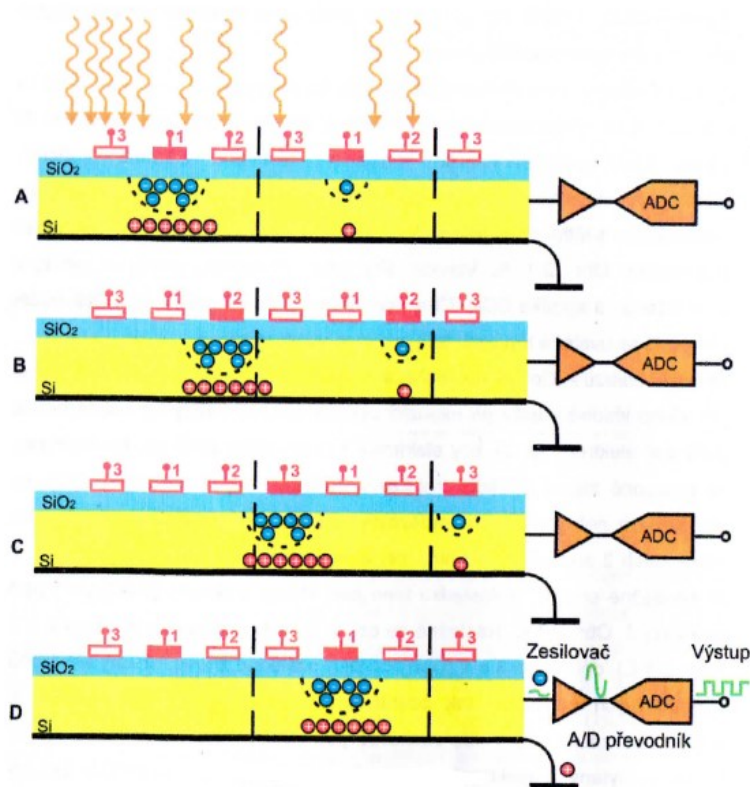
<i>Parametr</i>	<i>CCD</i>	<i>CMOS</i>
<i>Signál vystupující z pixelu</i>	paket elektronů	napětí
<i>Signál vystupující z čipu</i>	napětí (analogový signál)	bity (digitální signál)
<i>Rychlost</i>	nízká, sekvenční čtení	vysoká, maticově adresované buňky
<i>Citlivost</i>	velmi dobrá, lze dosáhnout kvalitního obrazu, lepší barevná věrnost snímků	za snížených světelných podmínek může být problém dosáhnout kvalitního zobrazení, horší rozlišení barev
<i>Dynamický rozsah</i>	vysoký	průměrný
<i>Šum</i>	malý, velká kvalita obrazu	je větší s ohledem na nižší fill factor a z toho vyplývající potřebu mikročoček
<i>Odběr energie</i>	odebírá více (uvádí se až 50x více než CMOS)	relativně malý
<i>Složitost čipu a náklady na vývoj</i>	menší	velké
<i>Složitost systému</i>	velká, mnoho obvodů je mimo čip	menší, většina obvodů je přímo na čipu
<i>Cena</i>	vysoká, jedná se o specializovanou výrobní technologii	levnější, je využíváno standardní technologie výroby logických obvodů

CCD

Základová destička snímače CCD (Charge-Couple Device, v překladu zařízení s vázanými náboji) je tvořena čistým křemíkem (Si), který funguje jako polovodič a je vespod uzemněn. Na druhé straně je tenká vrstva oxidu křemičitého (SiO_2), jež funguje jako dokonalý izolant.

Činnost CCD snímače je složena ze tří fází. Nejprve jsou z CCD odebrány všechny volné elektrony, to znamená, že je smazán zbytek předchozího nasnímaného obrazu. Následně je na elektrody přivedeno kladné napětí a je otevřena závěrka (na snímač působí světlo). Fotony viditelného spektra mají dostatek energie a uvolní elektrony z atomů tvořících křemíkovou strukturu. Počet uvolněných elektronů je úměrný množství dopadajícího světla. Čím více dopadajícího světla, tím více elektronů se uvolní a tím větší náboj vzniká. Konečně je snímán obraz, přičemž na každém čipu jsou vedle sebe umístěny tři elektrody a to tak, že elektroda 1 je mezi elektrodou 3 a 2. Za předpokladu, že kladné napětí je na elektrodách 1, je na všechny elektrody přiváděn trojfázový hodinový signál a to způsobí pozvolné zvyšování napětí na elektrodách 2. Současně s tímto procesem se snižuje napětí na elektrodách 1 a shluky elektronů jsou přitahovány pod elektrody 2. Tento proces se následně opakuje a tím se shluky elektronů z jednotlivých pixelů posouvají přes další pixely směrem k výstupnímu zesilovači. Tím je pak nízké napětí zesilováno dle odpovídající počtu nachytných elektronů v jednotlivých

pixelech na napětíové úrovni. Poté dochází pomocí A/D převodníku ke kvantování – převodu analogového signálu na signál digitální. Grafické znázornění principu činnosti snímače CCD na Obrázek 3.

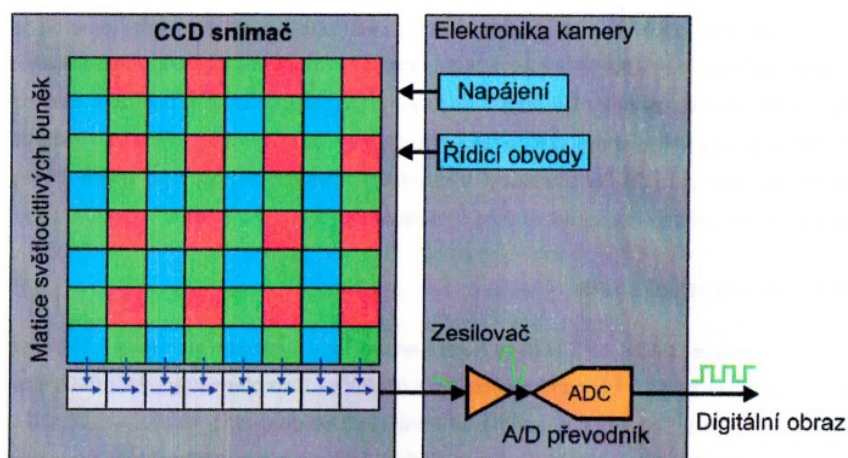


Obrázek 3: Princip činnosti CCD snímače [3]

[3]

Výše popsanou funkcí CCD snímače je vytvářen řádek pixelů – lineární informace. Toto je vhodné u lineárních kamer. U plošných senzorů je ovšem potřeba snímat dvojrozměrný obraz. Tyto senzory se tedy skládají z několika lineárních, které jsou naskládány na sebe a je tak vytvořena matice pixelů, viz Obrázek 4.

[3]

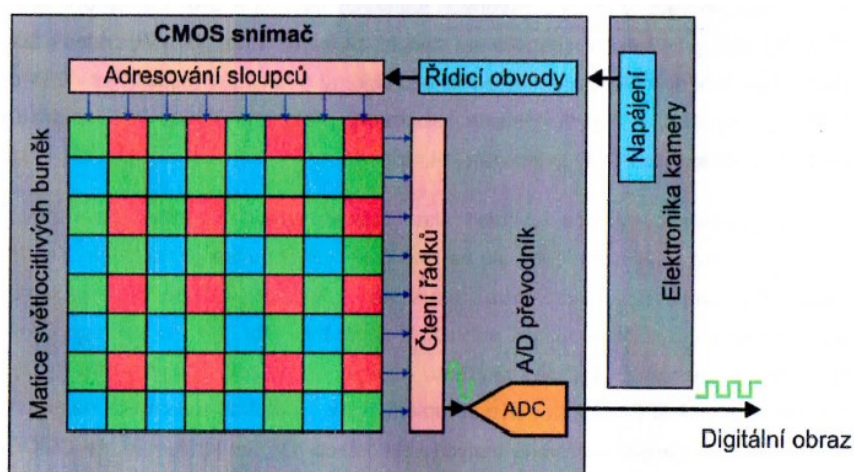


Obrázek 4: Architektura CCD snímače [3]

CMOS

Stejně jako u snímače CCD i v tomto případě je využíváno fotoefektu. Konstrukce je rovněž podobná, ale buňky CMOS snímače jsou čteny přímo. Každá světlocitlivá buňka čipu (každý pixel) má svůj vlastní zesilovač. Díky tomu je přímo adresována a čtena pomocí X, Y souřadnic onoho pixelu. Čtení obrazového záznamu je tím rychlejší, je také snižována spotřeba energie a produkce zbytkového tepla. Snímače CMOS jsou vyráběny v různých velikostech.

Výhodou těchto nastupujících snímačů je poměrně nízká cena a možnost umístění na čip všech potřebných prvků pro získání digitálního obrazu. Fakt, že každá buňka má vlastní zesilovač způsobuje, že na čipu je méně místa, a proto musí být samotná světlocitlivá fotodioda plošně menší, tím se snižuje fill factor¹. Tento nedostatek je řešen větším zesílením, to ovšem vede k většímu obrazovému šumu. Přidáním mikročočky nad každou světlocitlivou buňku je tomuto problému zabráňováno. [3]



Obrázek 5: Architektura CMOS snímače [3]

1.2.3 Světelná citlivost

Dalším důležitým parametrem je světelná citlivost kamery, respektive snímače. Udává, při jakých minimálních podmínkách je čip schopen snímat obraz. Předpokladem je, že čím více světla na snímanou scénu či objekt dopadá, tím je obraz kvalitnější. Oproti tomu při nedostatečném osvětlení je obraz zkreslený nebo příliš tmavý. Každá kamera má mez, při jejímž překročení již kamera nepracuje správně. Jednotkou intenzity osvětlení je lux [lx]. V Tabulka 2 porovnání určitých hodnot osvětlení vzhledem k prostředí. Čím nižší je hodnota osvětlení, tím lépe. Kupříkladu kamery s hodnotou 0,01 lx a nižší pracují dobře i v horších podmínkách, jako jsou nedostatečně osvětlená místa, špatné počasí nebo šero. Pro snímání ve večerních a nočních hodinách jsou lepší volbou černobílé kamery, jelikož mají vyšší citlivost a to je důležité při snížené úrovni osvětlení. [42]

¹ Fill factor – faktor plnění světlocitlivé buňky – poměr velikosti světlocitlivé plochy v buňce vzhledem k celé ploše buňky

Tabulka 2: Hodnoty osvětlení v různých prostředích [42]

<i>Prostředí</i>	<i>Osvětlení [lx]</i>
hvězdná obloha	0,00005
zatažená obloha bez měsíce	0,0001
srpek měsíce	0,01
úplněk a čistá obloha	0,1
svíčka ve vzdálenosti 30 cm	10
obývací pokoj	50
kancelář	150
západ a východ slunce	400
nákupní centrum	1 000
ranní slunce	4 000
odpolední slunce (min)	32 000
odpolední slunce (max)	100 000

1.3 Objektiv

Nezbytnou částí každé kamery je také objektiv, který má rovněž vliv na kvalitu obrazu. Samotný objektiv je složen soustavou několika čoček. U některých kamer jsou objektivы fixní částí kamery. Opačným případem jsou kamery s výměnným objektivem. Ten je nutno přizpůsobit prostředí, v němž je s kamerou pracováno. Objektivem je definováno, kolik scény bude snímáno a jak detailně. Dále je jím například kontrolováno množství světla, které prostupuje k obrazovému snímáči.

Vlastností objektivu je ohnisková vzdálenost, která říká, že se zvětšující se ohniskovou vzdáleností se zužuje pozorovací úhel a zároveň dochází k většímu přiblížení. Další vlastností je světelnost objektivu. V neposlední řadě důležitým parametrem je také velikost snímáče, ke kterému je dle této informace vybírán vhodný objektiv. Nejčastěji jsou používány senzory o velikosti 1/2, 1/3 či 1/4 palce. [39],[42]

1.4 Komunikační rozhraní

Přenos signálu mezi kamerou a záznamovým zařízením je obstaráván vybraným komunikačním rozhraním, neboli interfacem. Těchto rozhraní existuje celá řada a každý typ je vhodný pro určitý typ aplikací. Rozdíly mezi jednotlivými interfacemi jsou v rychlosti přenosu dat, ve zpoždění mezi pořízením snímku a jeho zobrazením, vzdálenosti, na kterou jsou data přenášena, způsobu napájení, zpoždění příkazů, zatížení procesoru a podobně. K základním typům rozhraní pro připojení kamer patří USB 2.0 a 3.0, FireWire (IEEE 1394 a/b), Gigabit Ethernet (GigE Vision) a Camera Link. V Tabulce 3 porovnání vlastností jednotlivých druhů rozhraní. [3]

Tabulka 3: Vlastnosti jednotlivých druhů rozhraní [3]

<i>Označení</i>	<i>USB 2.0</i>	<i>USB 3.0</i>	<i>IEEE 1394a</i>	<i>IEEE 1394b</i>	<i>GigE Vision</i>	<i>Camera Link</i>
<i>Maximální přenosová rychlost</i>	480 Mb/s	4800 Mb/s	400 Mb/s	800 Mb/s	1000 Mb/s	>2000 Mb/s (až 5440 Mb/s)
<i>Šířka pásma</i>	45 MB/s	600 MB/s	video: 32 MB/s (80%), celkově: 40 MB/s	video: 64 MB/s (80%), celkově: 80 MB/s	90 MB/s (120 MB/s pro sběrnici GigE)	255 MB/s (základní), 680 MB/s (plný)
<i>Maximální délka kabelu</i>	5 m	5 m	4,5 m	4,5 m	100 m	10 m
<i>Maximální délka při použití zesílení</i>	30 m	30 m	72 m	72 m	neomezeně	30 m
<i>Napájení kamery</i>	až 0,5 A, 5 V, přes USB	až 0,9 A, 5 V, přes USB	až 1,5 A, 36 V, z firewire kabelu	až 1,5 A, 36 V, z firewire kabelu	zvlášť (mimo s PoE)	zvlášť (mimo s PoCL)
<i>Podpora základní desky</i>	obecně všechny	relativně nové základní desky	mnoho	některé	některé	žádné
<i>Podpora OS</i>	Windows, Linux	Windows, Linux	Windows, Linux	Windows, Linux	Windows, Linux	v závislosti na dodavateli
<i>Zpoždění příkazu (PC - Kamera)</i>	-	-	-	350 – 500 μ s	600 – 900 μ s	do 1000 μ s
<i>Zpoždění příkazu (Kamera – PC)</i>	-	-	-	13 – 14 ms	13 – 14 ms	-
<i>Zatížení procesoru počítače</i>	střední (5 - 10%)	velmi nízké (méně než 2%)	velmi nízké (méně než 1%)	velmi nízké (méně než 1%)	střední (5 - 10%)	velmi nízké (méně než 1%)
<i>Připojení více kamer</i>		přes HUB	až 63 zařízení	4 porty na PCI kartě, možno použít HUB	teoreticky neomezeno	obtížně
<i>Podpora bezdrátového přenosu</i>	ne	ne	ne	ne	ano	ne
<i>Relativní cena</i>	nízká	nízká	středně vysoká	středně vysoká	nízká	vysoká

1.4.1 USB 2.0 a USB 3.0

Použití rozhraní USB 2.0 je vhodné pro menší vzdálenosti mezi počítačem a kamerou a také pro nižší datový tok, méně snímků za sekundu nebo nižší rozlišení. Pro připojení a napájení je použit jeden kabel. V průmyslových aplikacích je použití tohoto rozhraní spíše výjimka. Oproti tomu rozhraní USB 3.0 má mnohem lepší předpoklad pro využití při komunikaci s průmyslovými kamerami. Přenosová rychlost může být 10x vyšší než u USB 2.0, s čímž je také zvětšuje šířka pásma. Další zlepšení je také v možnosti napájení, a to až 0,9 A. Většina portů USB 3.0 umožňuje použití DMA – Direct Memory Access, to zajišťuje minimalizaci zátěže procesoru při přenosu velkých objemů dat. Aby bylo možno kamery používat i v systémech strojového vidění jako real-time zařízení, byl pro toto rozhraní vyvinut protokol USB3 Vision. [3]

1.4.2 IEEE 1394 a/b

Podobně jako USB rozhraní i toto komunikační rozhraní je vhodné především pro přenos dat na krátkou vzdálenost do 4,5 m. Maximální přenosová rychlost závisí na provedení, které je dnes nabízeno ve dvou standardech, a to Firewire 400 - typ „a“ a Firewire 800 - typ „b“. [3]

1.4.3 GigE Vision²

Rozhraní GigE je uplatňováno při připojení více kamer, případně na delší vzdálenost mezi počítačem a kamerou. Výhodou je, že po připojení se kamera chová jako standardní síťové zařízení. Pro komunikaci lze používat standardní ethernetovou kabeláž 5e. Možnost vzdáleného připojení přes VPN – Virtual Private Network nebo přes Wi-Fi. Nevýhodou tohoto rozhraní je vyšší ztížení procesoru počítače. Na rozdíl od předchozích rozhraní není napájení realizováno přímo z komunikačního kabelu a je potřeba přídavný zdroj. Výjimkou je tzv. PoE. Napájení kamery je řešeno přes kabely v ethernetu, na něž musí být přivedeno. [3]

1.4.4 Camera Link

Sběrnice Camera Link je používána pro přenos velkého rozlišení a při vysoké rychlosti snímání, pohybující se v řádech stovek snímků za sekundu. Nicméně nevýhodou Camera Linku je, že je nutno použít speciální kartu do PC a kabeláž je omezena na 10 m. [3]

² Standard GigE Vision definuje sadu komunikačních protokolů pro průmyslové kamery používající jako přenosové médium Ethernet.

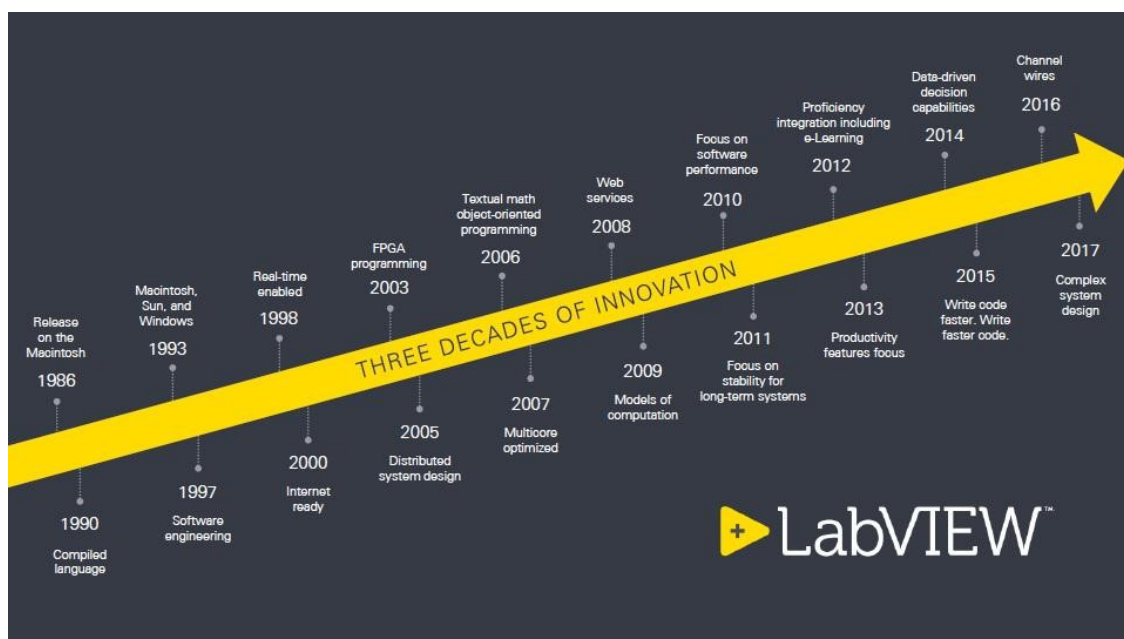
2 Vývojové prostředí LabVIEW

Kapitola 2 pojednává o vývojovém prostředí LabVIEW, od společnosti National Instrument, v něm bude vytvářena praktická část mé bakalářské práce.

2.1 Historie a vývoj

Vznik programovacího prostředí LabVIEW je řazen do roku 1983. Firma National Instruments (NI) uvedla na trh desku pro rozhraní GPIB. NI založil v roce 1976 James Truchard (přezdívaný Dr. T), James potřeboval vyřešit problém s připojením testovacího zařízení k minipočítači DEC PDP-11. Pro řešení tohoto problému přizval své kolegy Jeffrey Kodoskyho a Billa Nowlina, kteří pracovali na aplikacích sonaru pro americké námořnictvo na Texaské univerzitě v Austinu. Výsledkem jejich společného výzkumu byla myšlenka „virtuální instrumentace“³.

Cílem byla představa, aby technik, který je schopen zapsat své poznatky a požadavky do blokového diagramu, mohl takto lehce a intuitivně napsat i program. Vzniklo vývojové prostředí nejdříve pro počítače Apple Macintosh, které místo klasického textového programování umožňuje tvořit programy v grafické interpretaci. Program obsahoval různé funkce, reprezentované ikonkami, které lze vzájemně spojovat virtuálními vodiči a výsledky pak vykresloval např. do grafů. V té době to byla revoluční myšlenka. Grafický programovací jazyk, na kterém je založeno LabVIEW způsobil všeobecné rozšíření virtuální instrumentace a zpřístupnil automatizovanou instrumentaci všem technikům. [1]



Obrázek 6: Vývoj grafického prostředí LabVIEW [1]

³ Virtuální instrumentace – vytvoření virtuálního měřicího přístroje se všemi náležitostmi hardwarového přístroje

2.2 Princip a využití LabVIEW

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) je produktem společnosti National Instrument. Často je označováno jako otevřené grafické vývojové prostředí na bázi jazyka C. Toto pojmenování vyjadřuje otevřenost LabVIEW vzhledem k různým technologiím, které jsou v současnosti běžně používány. Hlavním účelem LabVIEW je tedy poskytnout nástroj, který umožní sjednocení nejrůznějších typů hardwaru a softwaru do jedné aplikace. Tato otevřenost umožňuje například ukládat data do různých typů databází a komunikovat s nejrůznějšími typy měřicích přístrojů jako jsou multimetry, osciloskopy, PLC apod. pomocí různých typů rozhraní (Ethernet, GPIB, CAN, RS-232 a další).

Základní myšlenkou LabVIEW je nahrazení klasických měřicích přístrojů za pomoci osobního počítače a převodníku. V každé z dnešních laboratoří nalezneme u všech měřicích přístrojů nějaký procesor, na němž běží aplikace, která definuje jeho funkcionalitu. National Instruments v tomto volí odlišný přístup. Pomocí LabVIEW je vytvořen „virtuální instrument“. Virtuální se označuje proto, že místo několika měřicích přístrojů je používán pouze jeden počítač s různými typy převodníků a komunikačních rozhraní. Softwarem je pak definovaná funkce konkrétního přístroje.

Program, jenž je v LabVIEW vytvářen se nazývá VI (z anglického Virtual Instrument). Každé VI se skládá ze dvou základních oken. Prvním z nich je „Front Panel“, který funguje jako uživatelské rozhraní a je tedy obdobou ovládacího panelu klasického přístroje. Nachází se na něm například grafy, posuvné řídicí prvky, indikační diody a další. Druhým oknem je „Block Diagram“, který je obdobou elektrických obvodů klasických měřicích přístrojů a zároveň definuje funkci každého přístroje.

Původní využití programu LabVIEW bylo za účelem měření, avšak jeho použití se rozšířilo do různých oblastí vývoje, výzkumu, testování a podobně. Zajímavým příkladem využití LabVIEW je například pro měření a řízení magnetů velkého kruhového urychlovače v největší evropské částicové laboratoři CERN.

[2]



Obrázek 7: LabVIEW 2018

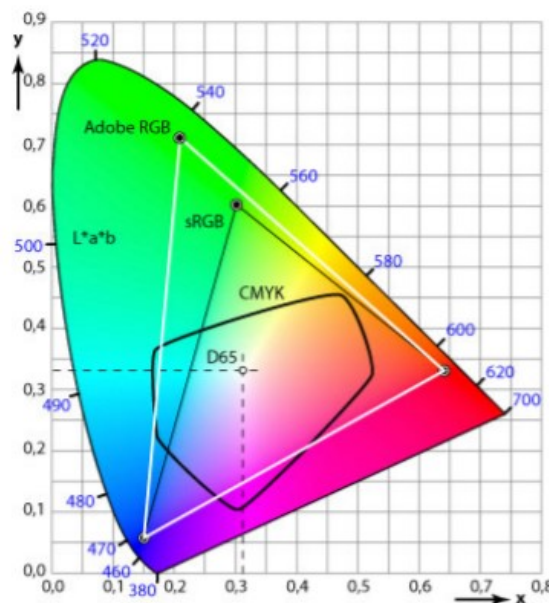
3 Základní pojmy a operace s obrazem

Tato kapitola se zabývá seznámením se se základními metodami při práci s obrazy, tedy s metodami, jež umožňují zlepšení kvality obrazu pro další zpracování. Výsledkem těchto metod může být redukce či odstranění šumu nebo jiných nedokonalostí způsobených nevhodnými podmínkami při pořizování obrazu – nepříznivé osvětlení, špatná ostrost obrazu (pohyb snímaného objektu, pohyb snímače) a podobně. Mezi metody používané při předzpracování se řadí zejména jasová a geometrická transformace, matematické morfologie a další. Před samotným předzpracováním obrazu je důležité si nejprve uvědomit, že snímaný obraz nebude ani pomocí nejmodernějších fotoaparátů zobrazen tak, jako je vnímán lidským okem, kde se neprovádí digitalizace obrazu. Jelikož u kamerových systémů je tato digitalizace přítomná, je jedna jednotka informace, taktéž nazvaná jako pixel, reprezentována v počítači pomocí specifické informace. Každý pixel má podle optické soustavy spočitatelnou velikost v reálném prostoru, a jeho hodnota odpovídá použitému barevnému modelu.

3.1 Jasová reprezentace obrazu

Uspořádání barev čili barevný model popisuje barvy dle podílu jednotlivých složek – základní barvy. V současnosti existuje mnoho barevných modelů, se kterými se můžeme setkat. My si zde popíšeme barevný model RGB, CMYK a HSV jakožto nejpoužívanější zástupce barevných modelů.

Každý barevný model má vlastní rozsah barev, které na něm mohou být zobrazeny. Tento rozsah je označován termínem gamut. K názornému vyobrazení gamutu je používán tzv. chromatický diagram, na němž jsou schematicky zobrazeny všechny barvy rozpoznatelné okem. V tomto diagramu každý model zaujímá různě velkou plochu, má odlišný tvar a pozici.



Obrázek 8: Chromatický diagram [11]

Na příklad u modelu RGB se předpokládá, že gamut bude konstantní s jasem, avšak u barevného modelu CMYK tomu tak není. Důsledkem těchto rozdílů je, že nastává problém při převodu z jednoho modelu na jiný. Pokud je nějaká barva mimo gamut určitého modelu, pak bude nahrazena nejbližší barvou, kterou prostor požadovaného modelu nabízí. Tímto procesem dochází k tzv. kompresi gamutu. [9], [11]

3.1.1 Model RGB

Model RGB je aditivní model, který vzniká na základě aditivního míchání barev⁴ a který vychází z faktu, že buňky v lidském oku jsou citlivé na vlnové délky, které odpovídají třem základním barvám – **R**ed (červená, vlnová délka 625 – 740 nm), **G**reen (zelená, vlnová délka 520 – 565 nm) a **B**lue (modrá, vlnová délka 430 – 500 nm). Kombinací těchto barev lze dosáhnout téměř jakékoli barvy.

Každá složka v RGB modelu je vyjádřena pomocí celého čísla od 0 do 255. Základní obraz je tvořen 24bity, na každou barvu tedy připadá 8 bitů, to lze zapsat jako 2^8 bitů = 1 byte = 256 hodnot. Barvy v tomto modelu lze také kódovat pomocí hexadecimálních čísel, kdy každá barva leží v rozsahu 0 – FF. Kódování vybraných barev v Tabulka 4.

Model lze znázornit pomocí jednotkové krychle, kdy jednotlivé osy odpovídají základním barvám, počátek odpovídá barvě černé a protilehlý bod pak barvě bílé, viz. Obrázek 9. Variantou RGB modelu je model RGBA, kde A představuje složku Alpha, která nese informaci o průhlednosti. Hodnota určuje poměr smíšení barvy s barvou pozadí. Při $A = 1$ dochází k neprůhlednosti, naopak při $A = 0$ je barva transparentní (průhledná).

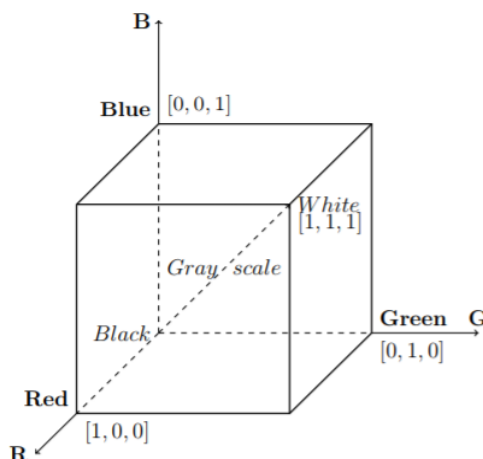
Využití barevného modelu RGB je četné. Nejčastěji pro fotoaparáty, televize, projektory, monitory a podobně. Model RGBA bývá využíván například rastrovým grafickým formátem PNG.

[9], [12], [13], [14], [17]

Tabulka 4: Kódování vybraných barev ve formě hexadecimální a dekadického kódu [14]

<i>Barva</i>	<i>Hexadecimální kód</i>	<i>Dekadický kód</i>
červená	FF0000	255, 0, 0
zelená	00FF00	0, 255, 0
modrá	0000FF	0, 0, 255
azurová	00FFFF	0, 255, 255
purpurová	FF00FF	255, 0, 255
žlutá	FFFF00	255, 255, 0
bílá	FFFFFF	255, 255, 255
černá	000000	0, 0, 0
světle modrá (např.)	9999FF	153, 153, 255
světle žlutá (např.)	FFFF99	255, 255, 153
šedá (např.)	505050	80, 80, 80

⁴ Aditivní míchání barev – sčítání jednotlivých barevných složek, kdy výsledkem je světlo největší intenzity



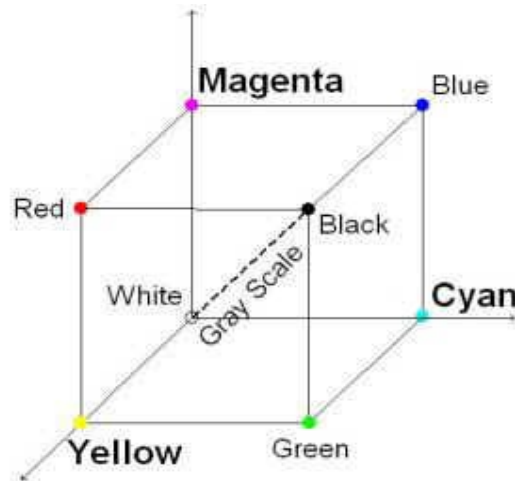
Obrázek 9 Model RGB [9]

3.1.2 Model CMYK

Model CMYK je na rozdíl od barevného modelu RGB založen na subtraktivním míchání barev⁵. Barevný prostor CMYK je tvořen 3mi základními barvami: Cyan – modrozelená, Magenta – fialová a Yellow – žlutá. Smícháním těchto barev, ale nikdy nevznikne černá, proto je přidána ještě poslední barva, black – černá. Nejčastější využitím tohoto modelu je v tiskárnách. To je dalším důvodem přidání černé barvy, černý toner bývá levnější a tisk jím je ostřejší.

Podíl jednotlivých barevných složek je stejně jako v případě RGB modelu udáván v hodnotách od 0 do 255 nebo v procentech v rozsahu od 0 do 100. I tento prostor můžeme popsat jednotkovou krychlí, z toho vyplývá, že převod mezi modelem RGB a CMYK je vcelku snadný. Výsledná barva vyjádřená v CMYK modelu je získána odečtením téže barvy vyjádřené v RGB modelu od jednotkové matice. Nicméně gamuty těchto modelů nejsou shodné, proto nemohou být všechny barvy modelu RGB zobrazeny v modelu CMYK a naopak. [9], [12], [13], [14], [17]

⁵ Subtraktivní míchání barev – přidáváním barev vzniká stále tmavší barva



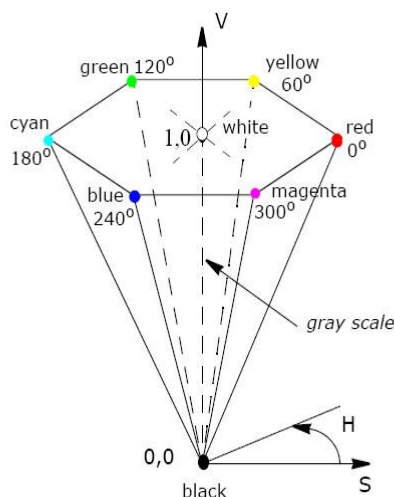
Obrázek 10: Model CMY [15]

3.1.3 Model HSV

Barevný model HSV používá pro definování barvy složky **Hue** – odstín, **Saturation** – sytost a **Value** – jasová hodnota, konkrétní barva je zde teda zobrazena již v první složce. Někdy bývá model HSV označován jako model HSB. Tento model má nejbližší k lidskému vnímání barev.

Na rozdíl od modelů RGB a CMYK je tento model zobrazován jako šestiboký jehlan umístěný do souřadnicového systému. Vrchol jehlanu se nachází v počátku souřadnicového systému a osa jehlanu je shodná se svislou osou. Odstín je měřen ve stupních, kde jednotlivé stupně představují převládající spektrální barvu. Na vodorovné ose je složka sytost, jež popisuje čistotu dané barvy a mění se v intervalu $<0;1>$ nebo je udávána v procentech od 0 do 100. Poslední složka, jasová hodnota, která udává, jak moc bílé barvy je v konkrétní barvě obsaženo. Hodnoty jasu se udávají v procentech a pohybují se v rozmezí 0 až 100. S rostoucí hodnotou se zvyšuje jas dané barvy.

Velkou výhodou tohoto modelu je možnost úpravy jasu a sytosti nezávisle na barevném tónu. Pro tuto svou vlastnost je model HSV využíván například pro editaci obrázků. Naopak nevýhodou modelu HSV je, že přechod mezi černou a bílou barvou není zcela plynulý. [9], [12], [13], [14], [17]



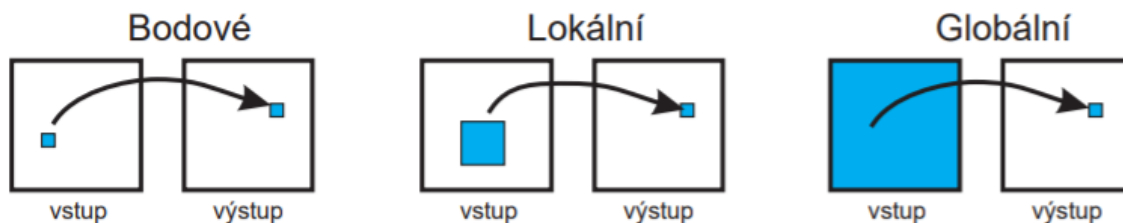
Obrázek 11: Model HSV a vyznačení jednotlivých složek a základních barev [15]

3.2 Jasové transformace obrazu

Jasová transformace je vratnou změnou hodnot obrazové funkce, která si klade za cíl zvýraznit určité rysy obrazu podle daného pravidla T . Vstupem i výstupem je obraz stejného rozlišení a stejné bitové hloubky, při transformaci nedochází k rozpoznání a interpretaci objektů. Při některých transformacích však může docházet i k nežádoucímu zvýraznění šumu. Jasové transformace se rozdělují primárně podle velikosti okolí vyšetřovaného bodu, a to na globální, lokální a bodové. [9], [18], [20]



Obrázek 12: Znárodnění jasové transformace [18]



Obrázek 13: Typy operací podle vlivu vstupního obrazu na výstupní obraz [19]

3.2.1 Globální jasová transformace

Globální operace složí k úpravě obrazu jako celku. Každý bod výstupního obrazu je u globálních operací ovlivněn všemi body vstupního obrazu. Patří sem například restaurační mechanizmy (odstranění zkreslení v obraze, potlačení šumu, aj.) nebo dvojrozměrné transformace obrazu (př. Fourierova transformace, kosinová transformace, aj.). Globální jasovou transformaci lze zapsat jako:

$$\Omega(I) \xrightarrow{T} f(x, y), \forall x, y \in \Omega(I); \quad (3.1)$$

kde symbol $\Omega(I)$ vyjadřuje doménu celého obrazu.

[4], [18]

3.2.2 Lokální jasová transformace

Lokální jasová transformace využívá ke změně jasu v daném pixelu pouze malé lokální okolí vyšetřovaného bodu, většinou to bývá nejbližší čtyřokolí nebo osmiokolí. Této transformace se využívá například při redukci šumu, zvýraznění hran či vyhlazování obrazu. Zápis lokální jasové transformace:

$$\Omega(x, y) \xrightarrow{T} f(x, y), \forall x, y \in \Omega(I); \quad (3.2)$$

kde symbol $\Omega(x, y)$ vyjadřuje blízké okolí daného bodu.

[9], [18], [20]

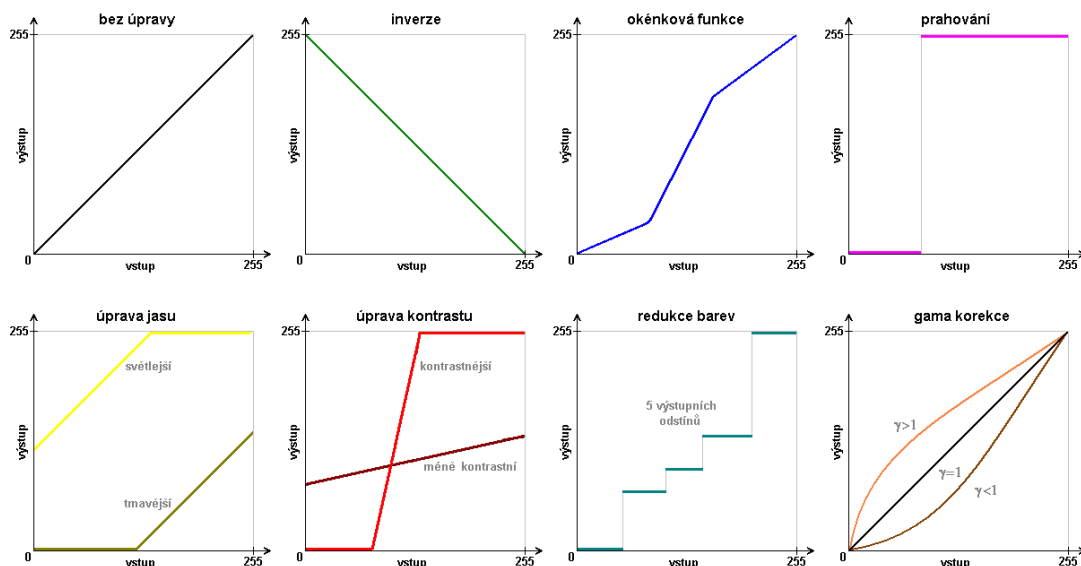
3.2.3 Bodová jasová transformace

Bodové operace slouží k transformaci obrazu bod po bodu, kdy každý bod výstupního obrazu je ovlivněn pouze jedním bodem vstupního obrazu. Tato transformace je z výše zmíněných ta nejpoužívanější, a to ve formě převodních charakteristik nebo tzv. LUT. Tyto tabulky udávají závislost jasových hodnot výstupního obrazu na jasových hodnotách obrazu vstupního. Lze pomocí nich provádět operace jako například prahování, úprava jasu a podobně.

Bodová jasová transformace lze zapsat jako:

$$f'(x, y) \xrightarrow{T} f(x, y), \forall x, y \in \Omega(I); \quad (3.3)$$

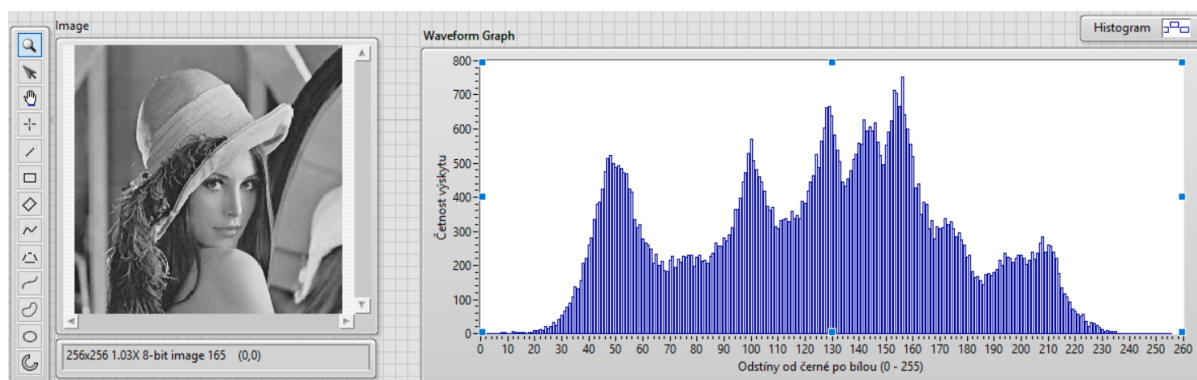
kde symbol $f'(x, y)$ znázorňuje hodnotu původní obrazové funkce.



Obrázek 14: Převodní charakteristiky [18]

Pro každou převodní funkci musí platit princip jednoznačnosti, pro každou vstupní hodnotu existuje právě jedna výstupní hodnota. Avšak více vstupních hodnot může být zobrazeno na jednu výstupní hodnotu. [4],[9], [18], [20]

Rozložení jasových úrovní v digitálním obraze lze popsat pomocí histogramu. Je to graf, na kterém je na ose x zobrazen jas od černé po bílou a na ose y hodnota intenzity jakou určitá barva vyzařuje. Příklad histogramu na Obrázek 15.



Obrázek 15: Ukázka histogramu vytvořeného v LabVIEW

Převod mezi RGB a černobílou fotkou je vcelku jednoduchý. Lidské oko vnímá každou barvu jinak. Například na modrou barvu je méně citlivé než na zelenou. Rovnice pro výsledný jas pak vypadá takto:

$$JAS = 0,3R + 0,59G + 0,11B \quad (3.4)$$

Na Obrázek 16 lze vidět, jak jsou některé barvy vnímány v přepočtu na černobílou barvu. [21]

Barva	R	G	B	Jas			
				Příspěvek z R	Příspěvek z G	Příspěvek z B	Celkem
Černá	0	0	0	0	0	0	0
Červená	255	0	0	$255 \cdot 0.3$	0	0	76
Zelená	0	255	0	0	$255 \cdot 0.59$	0	150
Modrá	0	0	255	0	0	$255 \cdot 0.11$	28
Červená+Zelená	255	255	0	$255 \cdot 0.3$	$255 \cdot 0.59$	0	227
Červená+Modrá	255	0	255	$255 \cdot 0.3$	0	$255 \cdot 0.11$	104
Zelená+Modrá	0	255	255	0	$255 \cdot 0.59$	$255 \cdot 0.11$	178
Střední šedá	127	127	127	$127 \cdot 0.3$	$127 \cdot 0.59$	$127 \cdot 0.11$	127
Bílá	255	255	255	$255 \cdot 0.3$	$255 \cdot 0.59$	$255 \cdot 0.11$	255

Obrázek 16: Relativní srovnání jasu pro lidské oko [21]

K jasovým transformacím se neodmyslitelně řadí roztažení a ekvalizace histogramu. Roztažení histogramu je početně jednoduchá operace, při níž dochází k tomu, že je nenulový rozsah histogramu obrazu normován na celý rozsah možných jasových hodnot. Výstupní obraz pak využívá všech jasových úrovní. Naopak oproti tomu ekvalizace neboli vyrovnaní histogramu je výpočetně náročnější transformace jasových hodnot využívající informaci o míře zastoupení jasových úrovní ve vstupním obrazu. Nejprve se vypočítává kumulativní histogram⁶ s prvkem n , který je dán součtem prvku $n-1$ a n prvku původního histogramu. Poté je vytvořena transformační funkce normováním kumulativního histogramu na rozsah jasových hodnot odpovídajících zvolenému výstupnímu obrazu. Nakonec se provádí samotná transformace jasových úrovní podle převodní charakteristiky. Technika ekvalizace se používá na šedotónový obraz pro úpravu jasové složky, konkrétně zvýraznění kontrastu. [9], [20]

3.3 Geometrická transformace obrazu

Jednou z nejčastějších operací užívaných při zpracování obrazu jsou právě geometrické transformace. Jedná se především o translaci (posunutí), rotaci, škálování (změnu měřítka) či zkosení. Další změny jsou realizovány skládáním výše zmíněných základních transformací.

Úpravy jsou prováděny na základě převodu kartézských souřadnic na souřadnice homogenní. Převod je popsán jako:

$$[X \ Y] \rightarrow [x \ y \ w], \quad (3.5)$$

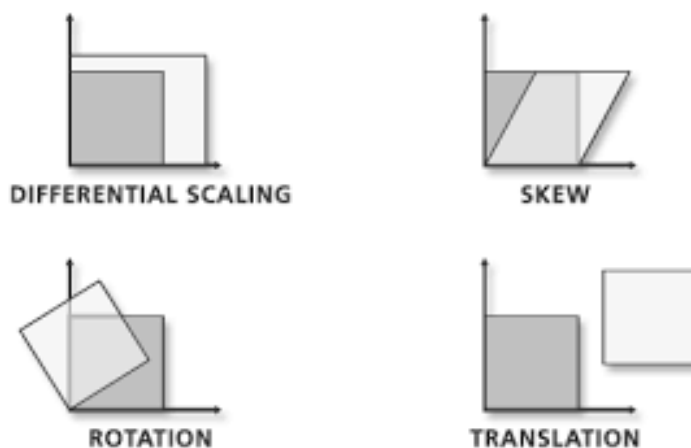
kde souřadnice w je váha bodu, která se využívá při modelování složitých těles.

⁶ Kumulativní histogram – integrace prostého histogramu

Pro běžné použití se však volí $w = 1$. Převod platí za předpokladu, že:

$$X = \frac{x}{w}, Y = \frac{y}{w}, w \neq 0. \quad (3.6)$$

[9], [22], [23]



Obrázek 17: Grafické znázornění jednotlivých geometrických transformací - změna měřítka, zkosení, rotace a posunutí [25]

3.3.1 Rotace

Při rotaci dochází k otočení objektu a všech jeho bodů o úhel α . Střed otáčení je počátek souřadnicového systému, například pro dvojrozměrný systém bod $[0, 0]$. Původní natočení objektu nehraje roli, předpokládá se, že objekt svírá s počátkem nulový úhel. Úhel α je libovolně volen dle toho, jak chceme objekt natočit. Podle znaménka určíme směr rotace, a to buď po směru hodinových ručiček (kladná rotace) anebo proti směru hodinových ručiček (záporná rotace). Následně je úhel α dosazen do transformační matice:

$$R(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.7)$$

Pro získání záporné rotace provedeme inverzi výsledné matice $R(\alpha)$.

[9], [23]

3.3.2 Změna měřítka

Změnou měřítka (angl. scale) je ovlivňována velikost obrazu, dochází tedy ke zmenšení nebo zvětšení obrazu. Prováděna operace se rozlišuje dle toho, jaké hodnoty nabývají koeficienty měřítka. Jestliže koeficienty nabývají hodnot z intervalu $(0; 1)$, pak dochází k posunu obrázku směrem k počátku souřadnicového systému, tedy ke zmenšení obrazu. Naopak pokud koeficienty měřítka nabývají hodnot z intervalu $(1; \infty)$ dochází k posunu směrem od počátku souřadnicového systému – ke zvětšení obrazu. Matice popisující 2D transformaci:

$$S(s_x, s_y) = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (3.8)$$

kde s_x je koeficient změny měřítka ve směru osy x a s_y je koeficient změny měřítka ve směru osy y . [9]

3.3.3 Posunutí

Translace čili posunutí je jednou z výpočetně nejjednodušších operací. Při této transformaci dochází k posunu každého pixelu obrazu o předem definovanou hodnotu, a to jak ve směru osy x , tak ve směru osy y . Matice pro posunutí se zapíše jako:

$$D(d_x, d_y) = \begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (3.9)$$

kde proměnné d_x a d_y určují hodnotu, o níž má být obraz posunut. [23], [24]

3.3.4 Zkosení

Zkosení (angl. shear transformation) se provádí podle osy x , kdy x' lineárně závisí na y . Pro tuto operaci slouží matice:

$$Sh_x(sh_x) = \begin{bmatrix} 1 & sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.10)$$

Pro zkosení podle osy y platí, že y' závisí na x a transformační matice vypadá:

$$Sh_y(sh_y) = \begin{bmatrix} 1 & 0 & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.11)$$

[9], [24]

3.3.5 Složené transformace

Složené transformace vznikají použitím výše zmíněných jednoduchých transformací v jednom kroku. Skládání se provádí pomocí násobení jednotlivých matic a jelikož násobení matic není komutativní operace, záleží na pořadí jednotlivých matic. Pokud tedy chceme obraz posunout a otočit, je nutné provést operace v tomto pořadí.

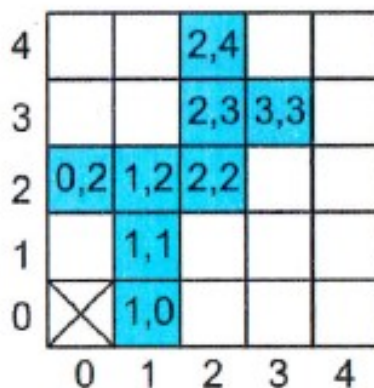
3.4 Matematické morfologie

Označení vychází z latinského morfe (tvar). Základní principy jsou založeny na nelineárních operacích v obraze a obraz je popsán za použití terminologie teorie množin, viz. Tabulka 5. Matematická morfologie využívá vlastností bodových množin, výsledků z integrální geometrie a topologie. Předpokladem pro tento druh operace je, že reálné obrazy lze modelovat pomocí množin libovolné dimenze. Běžným definičním oborem pro popis obrazu je dvojrozměrný euklidovský prostor E_2 . Při zpracování binárních obrazů použitím binární matematické morfologie je popisným nástrojem množina dvojic celých čísel $((x, y) \in \mathbb{Z}_2)$. Pro obrazy s více úrovněmi jasů se používá šedotónová matematická morfologie, pro níž je popisným nástrojem trojice celých čísel $((x, y, z) \in \mathbb{Z}_3)$. Matematická morfologie se používá při zpracování obrazu jako matematický nástroj pro výběr požadovaných součástí obrazu. Typická aplikace matematické morfologie je například při detekci hran. Původně byla matematická morfologie určena pouze pro analýzu binárních obrazů, později došlo k uzpůsobení i pro šedotónové obrazy. [4], [30]

Tabulka 5: Základní pojmy teorie množin [4]

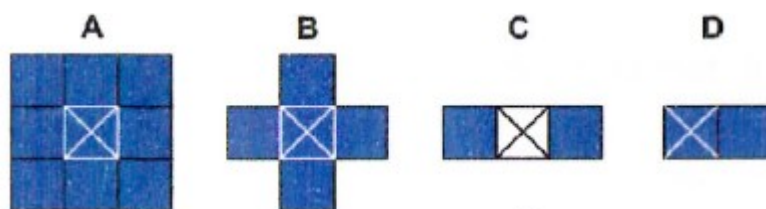
$\emptyset, \{\}$	PRÁZDNÁ MNOŽINA	označení množiny bez prvků	$\{n \in \mathbb{N} : 1 < n^2 < 4\} = \emptyset$
\subseteq	PODMNOŽINA	$X \subseteq Y$ značí, že každý prvek X je též prvkem Y	$X \cap Y \subseteq X$
\cup	SJEDNOCENÍ	$X \cup Y$ popisuje množinu, která obsahuje prvky, jež jsou alespoň v jedné z množin X a Y	$X \subseteq Y \Leftrightarrow (X \cup Y) = Y$
\cap	PRŮNIK	$X \cap Y$ popisuje množinu, která obsahuje prvky, jež jsou množinám X a Y společné	$\{x \in \mathbb{R} : x^2 = 1\} \cap \mathbb{N} = \{1\}$
\setminus	ROZDÍL	$X \setminus Y$ značí množinu, který obsahuje ty prvky X , jež neobsahuje Y	$\{1,2,3,4\} \setminus \{3,4,5,6\} = \{1,2\}$
\cup	SJEDNOCENÍ VŠECH PRVKŮ	$\cup X$ značí sjednocení všech prvků X	pro $X = \{b, c\}$ $b \cup c = \cup X = \cup \{b, c\}$
\cap	PRŮNIK VŠECH PRVKŮ	$\cap X$ značí průnik všech prvků X	
c	MNOŽINOVÝ DOPLŇEK	Y^c značí množinový doplněk	$X \setminus Y = X \cap Y^c$

Binární obrazy jsou takové, kdy obor hodnot je $\{0, 1\}$. Objekt nabývá hodnoty 1 a v obraze je reprezentován množinou U . Pozadí pak nabývá hodnoty 0 a tvoří doplněk U^C . Každý bod množiny = pixel (objekt nebo pozadí) je reprezentován dvojicí čísel, přičemž každé číslo popisuje polohu v diskretní mřížce v dané souřadné ose.



Obrázek 18: Příklad bodové množiny $U = \{(1,0), (1,1), (0,2), (1,2), (2,2), (2,3), (3,3), (2,4)\}$ [4]

Morfologické operace jsou realizovány jako relace obrazu (bodová množina U) a menší odlišnou bodovou množinou B , nazývanou strukturní element. Strukturní element B je vztažen k tzv. reprezentativnímu bodu (lokální počátek každého strukturního elementu – označen křížkem). [4]



Obrázek 19: Příklad strukturních elementů [4]

Z toho vychází morfologická transformace, kterou si lze představit jako určitý systematický pohyb strukturního elementu B po obrazu U a zápis do výsledného binárního obrazu. Základními matematickými morfologiemi jsou dilatace a eroze. Dalšími operátory, jež vycházejí z dilatace a eroze jsou pak otevření a uzavření nebo transformace tref a miň. K popisu těchto operátorů je využito tzv. Minkovských množinových operátorů. [4]

3.4.1 Dilatace

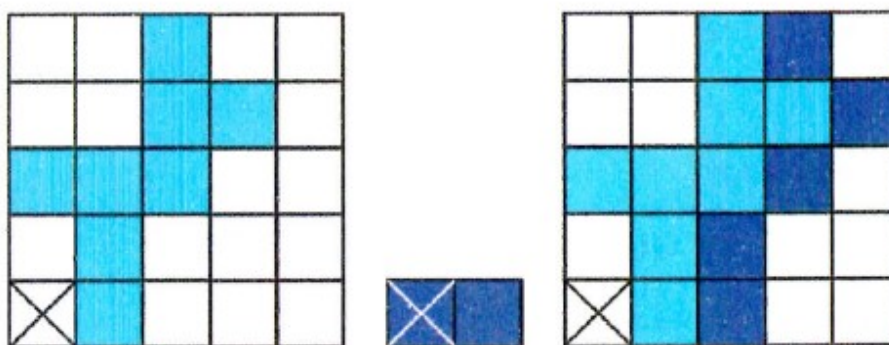
Dilatace je definována jako Minkovského množinový součet a značí se \oplus . Při dilataci dochází ke skládání dvou množin použitím vektorového součtu. Výsledkem dilatace $U \oplus B$ je bodová množina vektorových součinů pro dvojice pixelů, jeden z množiny U a druhý z množiny B . Popis dilatace pomocí vzorce:

$$U \oplus B = \{p \in \mathbb{E}_2 : p = u + b, u \in U, b \in B\}, \quad (3.12)$$

nebo jako sjednocení posunutých bodových množin

$$U \oplus B = \bigcup_{b \in B} U_b. \quad (3.13)$$

[4]



Obrázek 20: Příklad dilatace (vlevo původní množina U , uprostřed množina B , vpravo výsledná množina po dilataci $U \oplus B$) [4]

3.4.2 Eroze

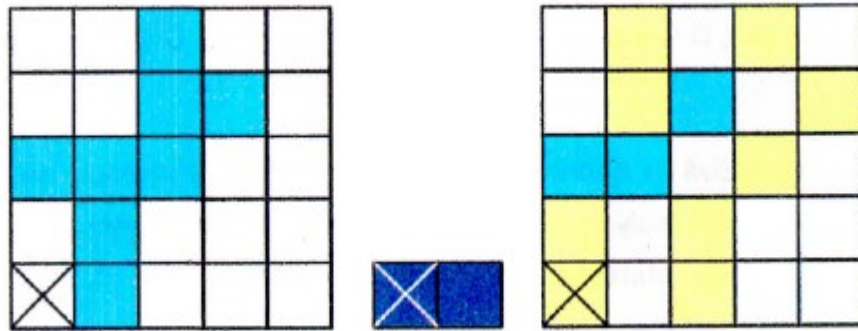
Eroze je definována jako Minkowského rozdíl s označením \ominus . Podstatou této operace je, že se pro každý bod obrazu p ověří, zda pro všechna možná $p+b$ leží výsledek v U . Pokud $p+b \in U$, výsledkem je 1. V opačném případě je do reprezentativního bodu zapsána 0. Eroze je popsána rovnicí:

$$U \ominus B = \{p \in \mathbb{E}_2 : p + b \in U \text{ pro každé } b \in B\}, \quad (3.14)$$

Nebo jako průnik všech posunů obrazů U o vektor $-b \in B$:

$$U \ominus B = \bigcap_{b \in B} U_{-b}. \quad (3.15)$$

[4]



Obrázek 21: Příklad eroze (vlevo původní množina U , uprostřed množina B , vpravo výsledná množina po erozi $U \ominus B$) [4]

3.5 Segmentace obrazu

Segmentace obrazu je jedna z nejdůležitějších částí při zpracování obrazu. Pojem „segmentace“ v analýze obrazu znamená, že vstupní obraz je rozdělen na části, které souvisí s předměty reálného světa. Při segmentaci dochází k oddělení objektů od pozadí. Každému bodu (pixelu) je přiřazen index segmentu představující určitý objekt v obraze. Cílem segmentace je většinou právě nalezení konkrétního objektu a jeho oddělení. Existuje několik technik, které mohou být při segmentaci obrazu použity. Níže uvedené techniky patří mezi nejčastěji využívané. [32],[33],[34]

3.5.1 Prahování

Prahování (angl. thresholding) je nejstarší a zároveň nejjednodušší segmentační metodou. Princip prahování je založen na myšlence, že objekt a pozadí má různou úroveň intenzity. Přičemž je definován práh $= T$, s nímž je každý pixel porovnáván. Pokud je hodnota pixelu menší než zadaný práh, je určen jako pixel pozadí. Naopak, má-li pixel větší hodnotu než zadaný práh, jedná se o pixel objektu. Výsledek prahování je určen po jednom průchodu obrazu. Níže na Obrázek 22 je příklad prahování v LabVIEW. Práh je nastaven na hodnotu $T = 128$. Všem pixelům majícím menší hodnotu, než je tento práh je přiřazena nulová hodnota a všem pixelům s větším hodnotou, než T je přiřazena hodnota maximální, tedy 255. [31],[32],[33],[34]



Obrázek 22: Příklad prahování

3.5.2 Detekce hran

Na principu detekce hran jsou založeny některé algoritmy, jež jsou využity při zpracování obrazu. Definice hrany je udávána jako skoková změna jasové funkce. Klasickými metodami používanými při detekci hran jsou metody založeny na první či druhé parciální derivaci.

Podstatou metody založené na první parciální derivaci je to, že k největší změně intenzity dochází v místě hrany, kdežto v homogenních oblastech reprezentujících pozadí je změna intenzity a současně s tím i první derivace rovna 0. Prvními parciálními derivacemi obrazu podle x a y je zjištěn vektor, který udává směr a velikost gradientu. Gradient funkce $\nabla f(x, y)$ je dán jako vektor, který je kolmý na vektor udávající směr hrany:

$$\nabla f(x, y) = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}, \quad (3.16)$$

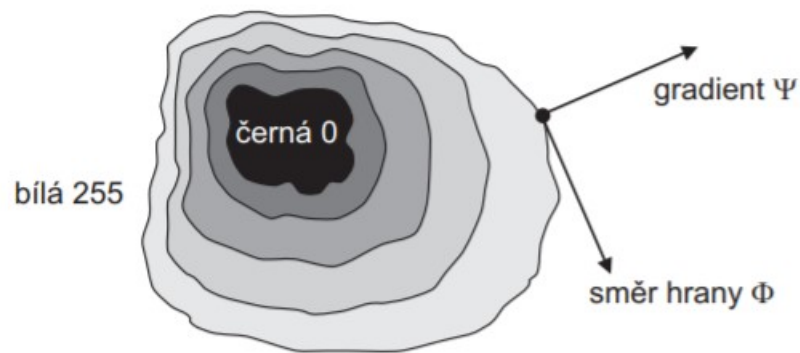
kde G_x a G_y jsou parciální derivace obrazové funkce $f(x, y)$ dle osy x , respektive osy y . Velikost a směr gradientu se pak vypočítá jako:

$$|\nabla f(x, y)| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}, \quad (3.17)$$

$$\psi(x, y) = \arg\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right), \quad (3.18)$$

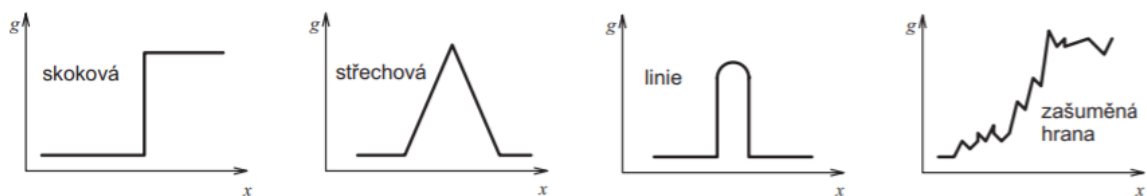
kdy $|\nabla f(x, y)|$ je velikost gradientu a $\psi(x, y)$ je směr gradientu.

[26],[27],[28],[29]



Obrázek 23: Grafická definice gradientu [26]

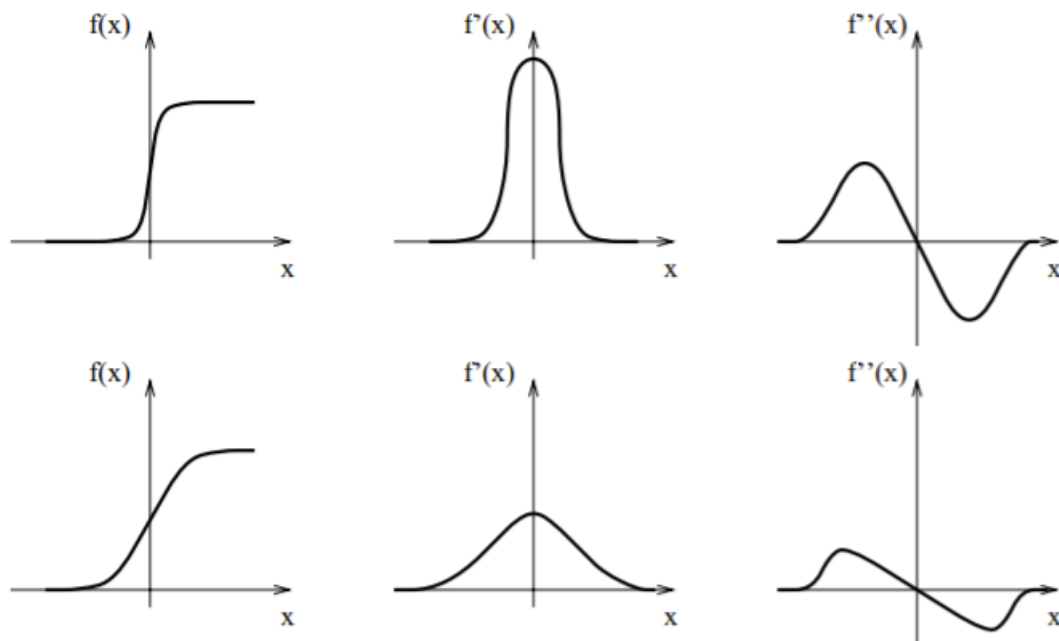
Kromě skutečných hran se v obraze objevuje také nenulová míra šumu. Tento šum je pro další zpracování nežádoucí, proto je třeba jej eliminovat. K tomu se nejčastěji využívá tzv. Gaussova filtru nebo využitím mediánu.



Obrázek 24: Znáznornění některých hran: vlevo skoková hrana, považována za ideální; vpravo zašuměná hrana, již lze nalézt v reálném obraze [26]

Po odstranění šumu jsou vypočítávány jednotlivé složky gradientu za použití jednoho z hranových operátorů. Nejčastěji používaný operátor o velikosti 2×2 je Robertsův operátor. Operátory o velikosti 3×3 jsou pak Prewittův, Sobelův, Robinsonův a Kirschův.

Druhou zmíněnou metodou je metoda založená na druhé parciální derivaci. Na rozdíl od gradientní metody, jak někdy bývá označována první metoda, v tomto případě se nemusí znát velikost a směr hrany. V místě největší změny intenzity a zároveň maximální první derivace se druhá derivace nachází v 0.



Obrázek 25: Vlevo původní funkce, uprostřed první derivace, vpravo druhá derivace [26]

Pro nalezení hrany je třeba zjistit druhou derivaci obrazové funkce a poté místa, kde funkce prochází 0. K tomu se používá nejčastěji Laplaceových operátorů ve čtyř nebo osmiokolích. Důraz těchto operátorů je kladen na středové body a součet prvků ať už čtyřokolí, tak i osmiokolí je roven 0. [26],[27],[28],[29]

4 Připojení kamery do LabVIEW

Kapitola 4 poskytuje nejprve několik informací o kameře, jež je použita při praktické části této bakalářské práce. Druhou podkapitolou je pak krátká zmínka o softwaru, pomocí něhož se nejprve otestuje funkčnost kamer.

4.1 Webová kamera LifeCam Studio Q2F-00018

Pro realizaci praktické části bakalářské práce byla vybrána webová kamera LifeCam Studio od společnosti Microsoft. Parametry kamery jsou následující: délka – 4.45“/113 mm a šířka – 1.57“/40 mm. Předností této konkrétní kamery je možnost otáčení se jak doprava, tak doleva o 180° a zachytit tak v případě potřeby celý okolní prostor. Pro mou práci nicméně tato funkce zapotřebí není, jelikož jsou kamery fixovány v jedné pozici. Zorný úhel v takové pozici je 75°. Počet snímků za sekundu lze nastavit manuálně, a to buď 5 fps nebo 30 fps (frames per second). Tohoto čísla ale prakticky není dosaženo nikdy. Reálný počet snímků, jež jsou vykreslovány, je závislý na vytížení procesoru. Kamera je vybavena snímačem s rozlišením 1080p⁷. Typ snímače použitý v této kameře je CMOS. To může mít vliv na rozlišení barev. Dále je kamera vybavena funkcí automatického zaostřování, a to od 0,1 m až do 10 m. [7],[44]

K uskutečnění této práce jsou použity 3 výše zmíněné kamery LifeCam studio, to je také jedním z aspektů, proč byla vybrána právě tato kamera. Cenová relace se pohybuje v přijatelné výši vzhledem k potřebnému množství, a především vzhledem k technickým parametrům kamery.

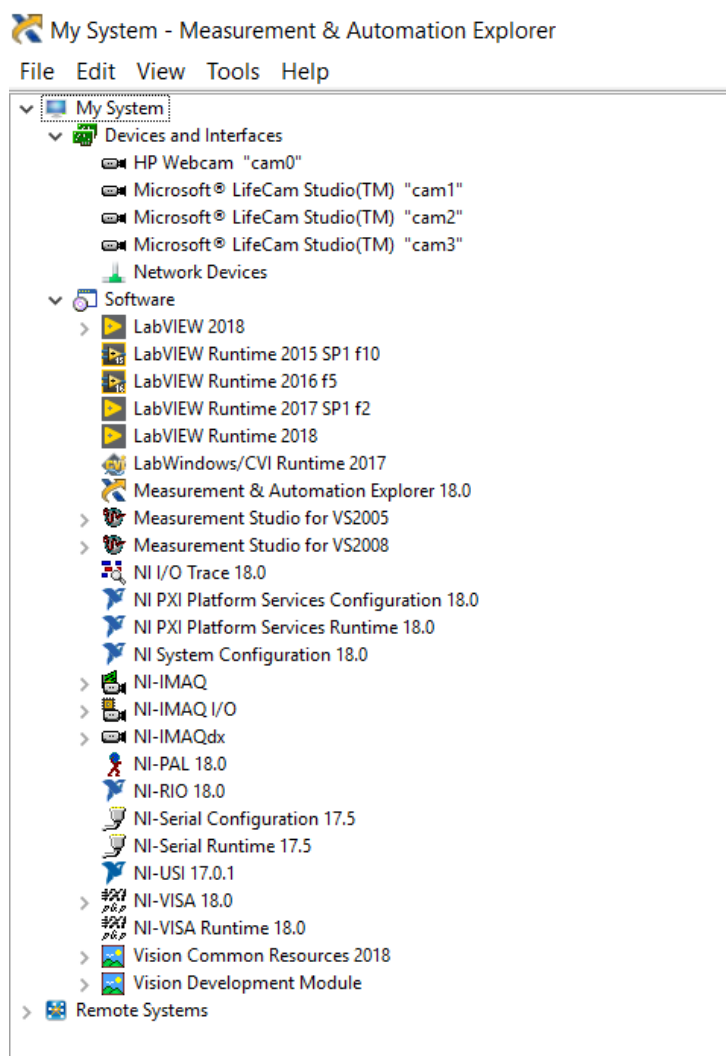


Obrázek 26: Microsoft LifeCam Studio Q2F - 00018 [8]

⁷ 1080p – 1080 značí počet obrazových bodů, pixelů, na výšku obrazu; p značí, že obraz je tzv. progressive – neprokládaný, tedy vytvářen postupně

4.2 Komunikace webové kamery a LabVIEW

Kamery jsou do počítače připojeny přes USB port. Zdařilé připojení je ověřeno v tzv. NI MAX. NI's Measurement & Automation Explorer je software, který se nainstaluje automaticky při instalaci programovacího prostředí LabVIEW. Tento software umožňuje zobrazovat a spravovat přídavné drivery. Pro zpracování obrazu a práci s kamerami je zapotřebí IMAQ driver, který je nutno doinstalovat. Pokud je vše správně nainstalováno, po připojení kamer/kamery lze v NI MAX sledovat a parametrovat obraz v reálném čase a současně je to ujištění, že při správné implementaci kódu, bude kýžený obraz rovněž výsledkem i v LabVIEW. Na Obrázek 27 lze vidět načtené kamery v NI MAX a rovněž doinstalovaný IMAQ driver.



Obrázek 27: NI MAX

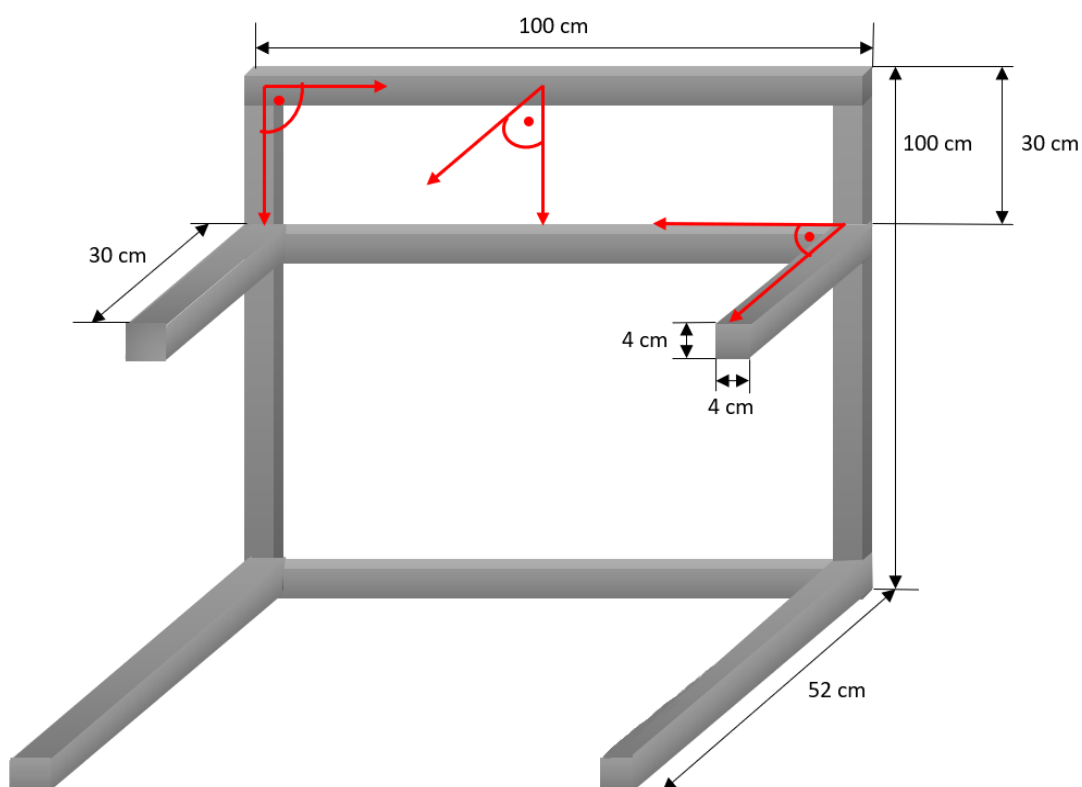
Vzhledem k faktu, že jsou kamery 3, mohly být připojeny do notebooku bez jakékoliv redukce. Komunikace je zajištěna přes USB 2.0. To znamená, že maximální rychlost činí 480 Mbit/s a je kompatibilní s USB 1.1 (přenosová rychlost buď 1,5 Mbit/s nebo 12 Mbit/s).

5 Pracovní prostředí

Jako pracovní prostředí, v němž bude objekt detekován a jež bude snímán kamerami byla vybrána hliníková konstrukce. Podrobné parametry viz. Obrázek 28 a Obrázek 29. Následně bylo třeba vybrat objekt, který bude detekován.

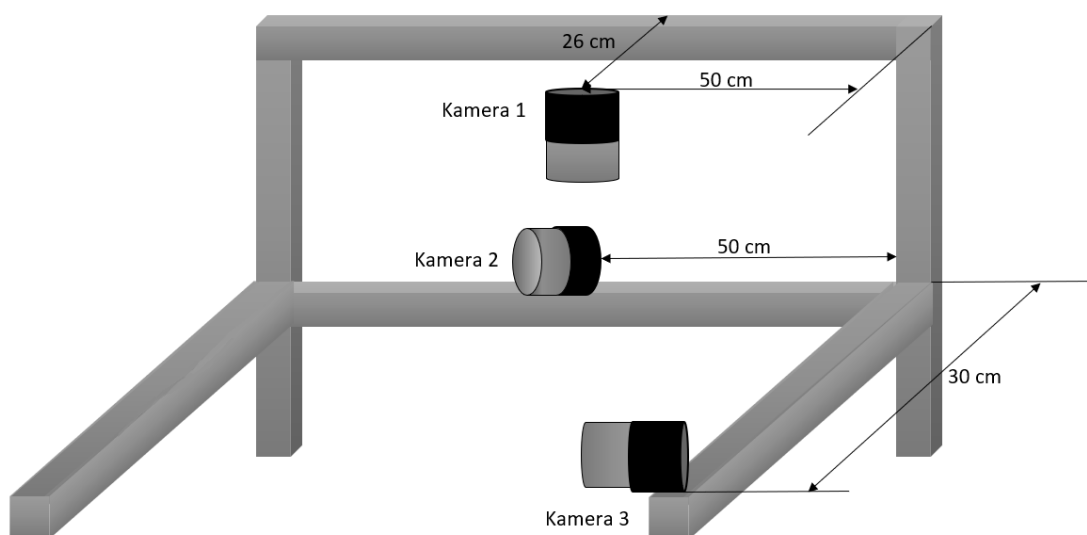
5. 1 Snímaný prostor

Jak jsem již zmínila výše, pro realizaci mé práce jsou využity 3 kamery. Ty jsou upevněny na hliníkovou konstrukci, která byla již připravena a nemohla jsem ji jakkoliv ovlivnit. Samotné rozmístění kamer bylo provedeno tak, že vůči sobě svírají úhel 90° , viz. Obrázek 28.



Obrázek 28: Parametry konstrukce

Kamery jsou na konstrukci připevněny pomocí speciálních držáků. Rozmístění kamer bylo zvoleno takovým způsobem, aby mohl být bez problémů snímán vyhrazený prostor. První kamera je umístěna 14 cm nad konstrukcí tak, aby snímala půdorys. Druhá kamera je umístěna na zadním dílu a je jí snímán nárys a poslední, třetí kamera je umístěna na bočním dílu a snímá půdorys.



Obrázek 29: Rozmístění kamer na konstrukci

Původním záměrem bylo vyhledávat zvolený objekt, na nekonstantním pozadí. Nicméně při následném testování jsem zjistila, že objekt byl detekován nepřesně nebo nebyl detekován vůbec, protože se v prostoru nacházelo něco výraznějšího. Podrobnější popis viz .5. 2 Detekovaný objekt. Na základě tohoto zjištění jsem se rozhodla snímaný prostor „uzavřít“ a vytvořit konstantní pozadí černé barvy.

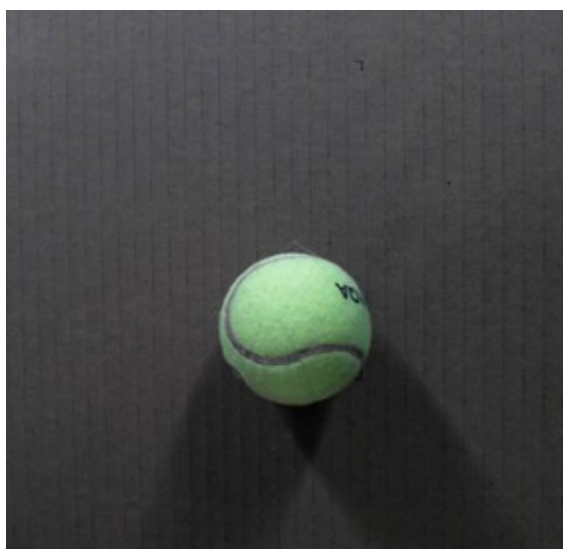


Obrázek 30: Reálný prostor

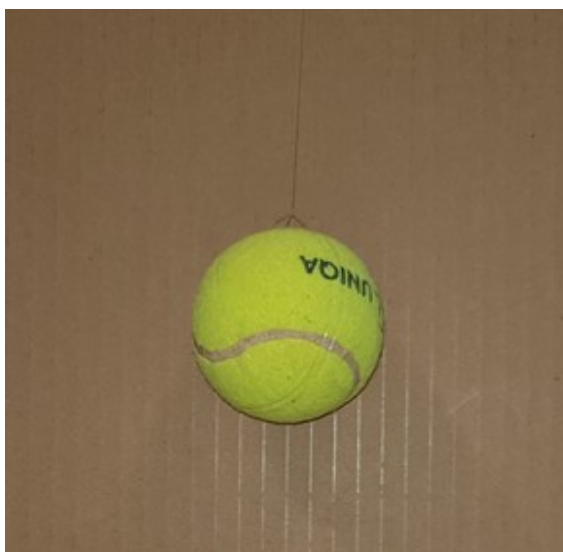
5. 2 Detekovaný objekt

Nezbytnou součástí mé práce je rovněž objekt, který bude detekován. Zkusila jsem několik objektů a nakonec jsem se rozhodla pro tenisový míček o průměru 65 mm, a to hned ze dvou důvodů. Prvním a hlavním důvodem je nezaměnitelná žlutozelená barva tenisového míčku. Druhým důvodem je pak přijatelná velikost vzhledem k velikosti prostoru a také tvar.

Jak jsem již zmínila, i přes výraznou a okem dobře rozpoznatelnou barvu míčku docházelo při testování s nekonstantním pozadím k nepřesné detekci. Při zjišťování, proč tomu tak je, jsem si nejprve zobrazila barevný obrázek z LabVIEW – Obrázek 31, následně jsem pro porovnání stejnou scénérii zaznamenala ještě mobilním telefonem. – Obrázek 32. Ani toto zobrazení není takové, jak je míček viděn lidským okem, nicméně barevné rozlišení je viditelně kvalitnější.



Obrázek 31: Barevný obrázek uložen z LabVIEW



Obrázek 32: Barevný obrázek pořízen mobilním telefonem

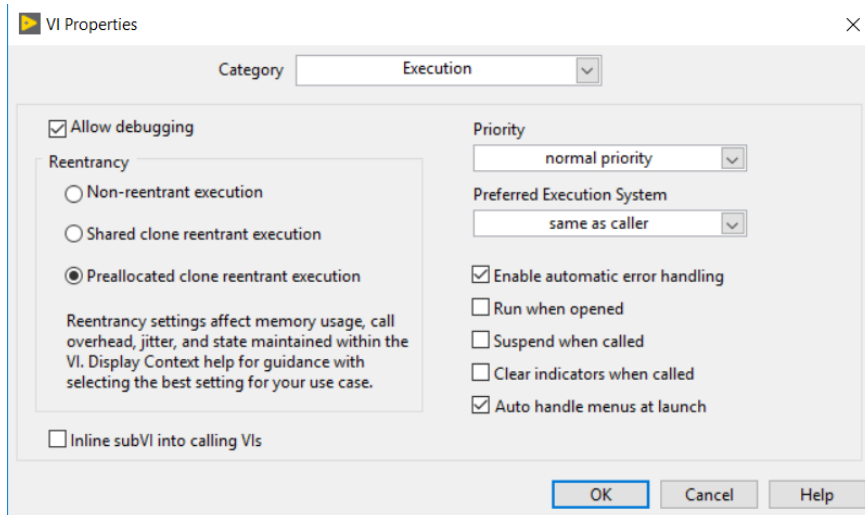
6 Aplikace v LabVIEW

Cílem mé bakalářské práce bylo vytvoření aplikace v programovacím prostředí LabVIEW pro snímání, detekování a vykreslování objektu v reálném čase pomocí více-kamerového systému. V následující kapitole jsou nejprve rozebrána jednotlivá subVI a následně vysvětlena funkčnost programu.

6.1 Rozbor jednotlivých částí

Podstatnou částí mé aplikace jsou subVI. V podstatě to není nic jiného, než určitá část kódu zapouzdřená do jedné „kostičky“. To se provádí především k zprehlednění blokového diagramu a rovněž proto, že některé operace se v programu opakují a musely by být vkládány stále znovu. V tu chvíli přichází na řadu subVI, kdy je ona operace do kódu vložena v této formě.

Důležitým aspektem, který je třeba ohlídat, je tzv. reentrantnost. Každé nové subVI má defaultně nastavenou ne-reentrantnost. To znamená, že toto subVI může běžet nejvýše jednou, jelikož má pouze jeden paměťový prostor a při vícenásobném použití tohoto subVI by se data přepisovala. Pokud tedy mají být spuštěny dvě nebo více instancí jednoho subVI, musí se tato vlastnost povolit, viz Obrázek 33. V takovém případě pracuje každý klon na svém paměťovém prostoru.

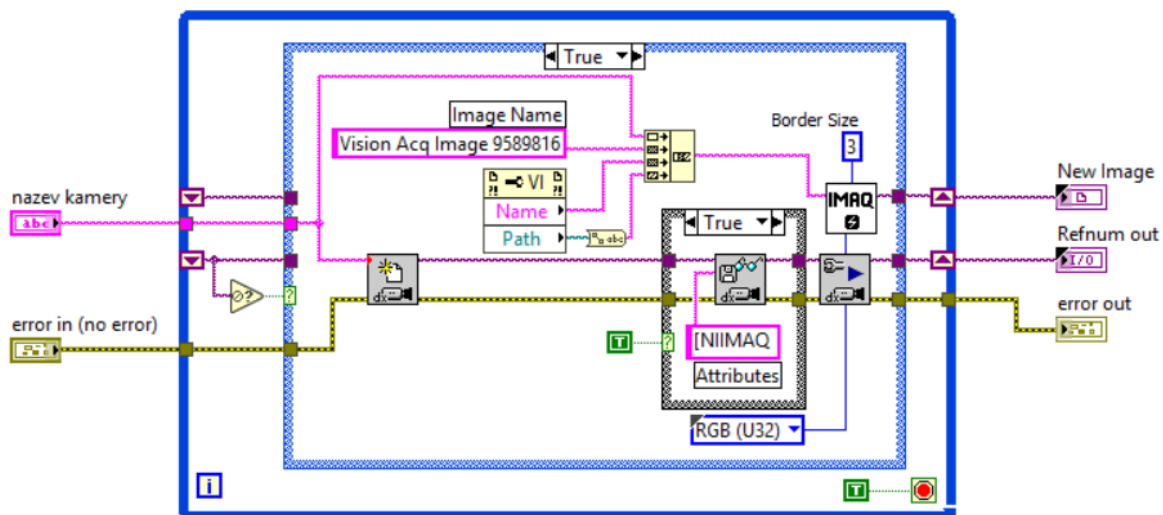


Obrázek 33: Nastavení reentrantnosti

Jelikož při své práci využívám 3 kamery a každá má svůj kód, jsou rovněž i níže zmíněná subVI využita 3x. Jediným subVI, které je v aplikaci použito pouze jednou, je pro vykreslování hodnot do grafu.

6.1.1 Načtení kamery

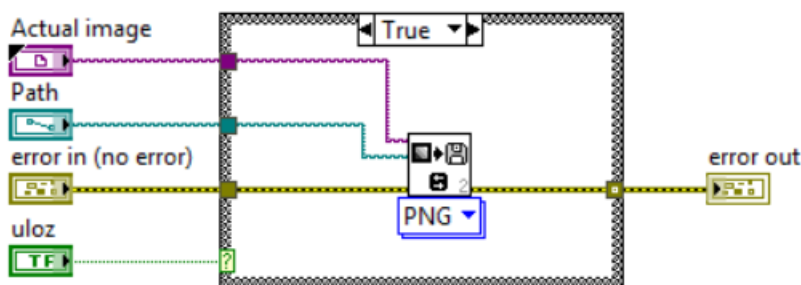
Prvním subVI, jenž jsem ve své aplikaci použila, bylo pro načtení kamery do programu. Dále je zde nastavován paměťový prostor pro snímání obrazu a rovněž je nastavováno v jakém barevném formátu bude obraz vykreslován.



Obrázek 34: SubVI pro načtení kamery

6.1.2 Sejmутí referenčního pozadí

Druhé subVI zajišťuje uložení aktuálního snímku do složky, kde je umístěn program. Tento snímek je definován jako pozadí. Toto subVI je spouštěno stiskem tlačítka *ULOZ* a je možné ho použít kdykoliv za běhu programu. Snímky jsou ukládány ve formátu PNG, ale lze si zvolit i jiný formát. Rozměry snímku jsou 1920x1080 pixelů a jejich bitová hloubka⁸ je 8. Takový obraz má 2^8 tedy 256 možných hodnot. Jelikož jsou snímky ukládány v monochromatickém provedení, těchto 256 (0 – 255) možných hodnot je ve stupních šedi.

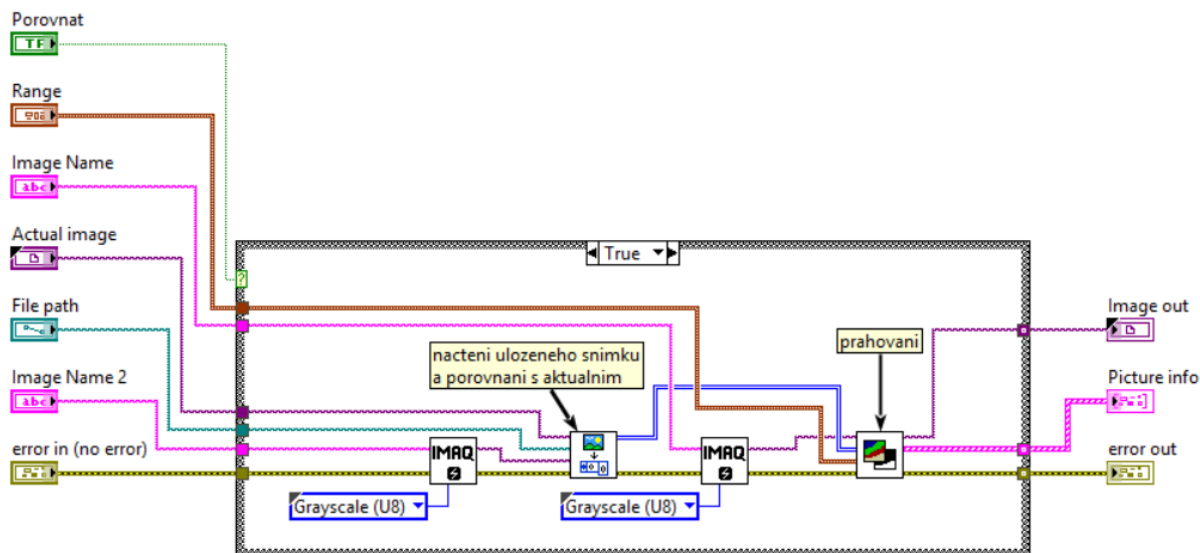


Obrázek 35: SubVI pro sejmутí referenčního pozadí

⁸ Bitová hloubka – určuje, kolik barevné informace je dostupné pro každý pixel

6.1.3 Porovnání

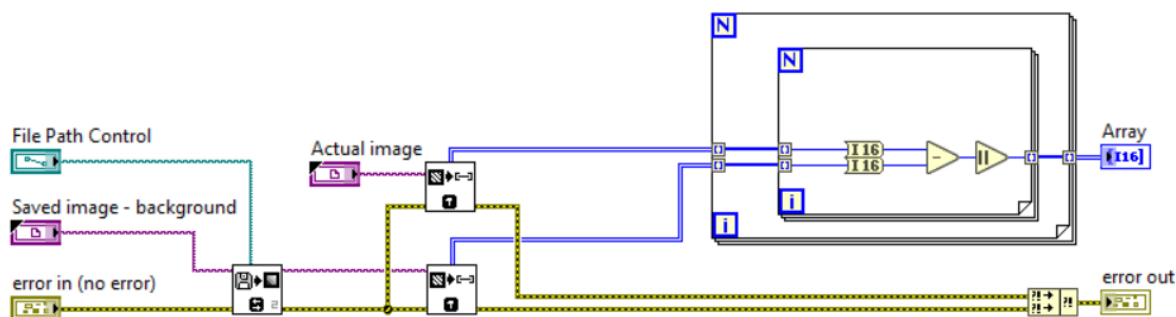
Toto subVI má podobně jako subVI pro ukládání obrázku na svém vstupu tlačítko. Stiskem tlačítka *POROVNAT* tedy dochází ke spuštění tohoto subVI. Uvnitř jsou vytvořeny další 2 subVI, jež jsou vysvětleny níže. „Kostičky“ *IMAQ* vytvářejí v paměti místo pro nové obrázky.



Obrázek 36: Rozdíl aktuálního snímku a uloženého snímku a následná segmentace

Převod na pole

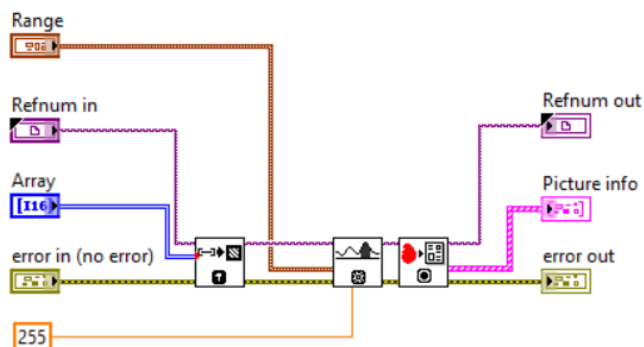
V tomto subVI je brán aktuální snímek a zároveň je načítán uložený snímek z paměti. Oba obrázky jsou převáděny na pole a následně od sebe odečteny. Aby nedocházelo k záporným hodnotám ve výsledku, je použita absolutní hodnota.



Obrázek 37: Převod obrázků na pole

Prahování

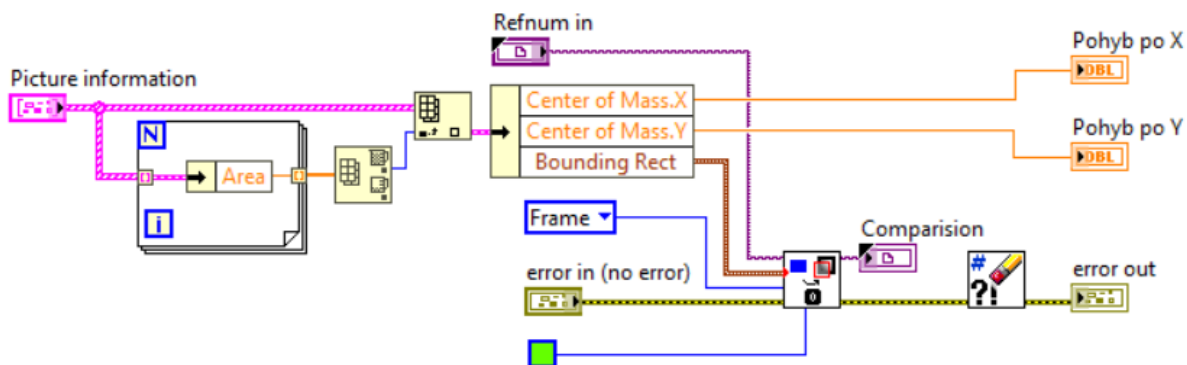
Výsledné pole je v tomto subVI převáděno zpět na obraz s dochází k prahování. Na čelním panelu celé aplikace je možnost pomocí slideru – ovládacího prvku nastavit prahovací rozsah. Pixelům, které leží v tomto rozsahu je pak nastavena hodnota 255, což je bílá a tyto pixely tvoří hledaný objekt. Všechny ostatní pixely, které nenáleží zvolenému rozsahu jsou vyhodnoceny jako pozadí a je jim nastavená hodnota 0, tedy černá barva. Na výstupu jsou informace o výsledném obrazu, z nichž jsou pak vybírány například souřadnice hledaného objektu a s nimi dále pracováno.



Obrázek 38: Prahování

6. 1. 4 Vyznačení hledaného objektu

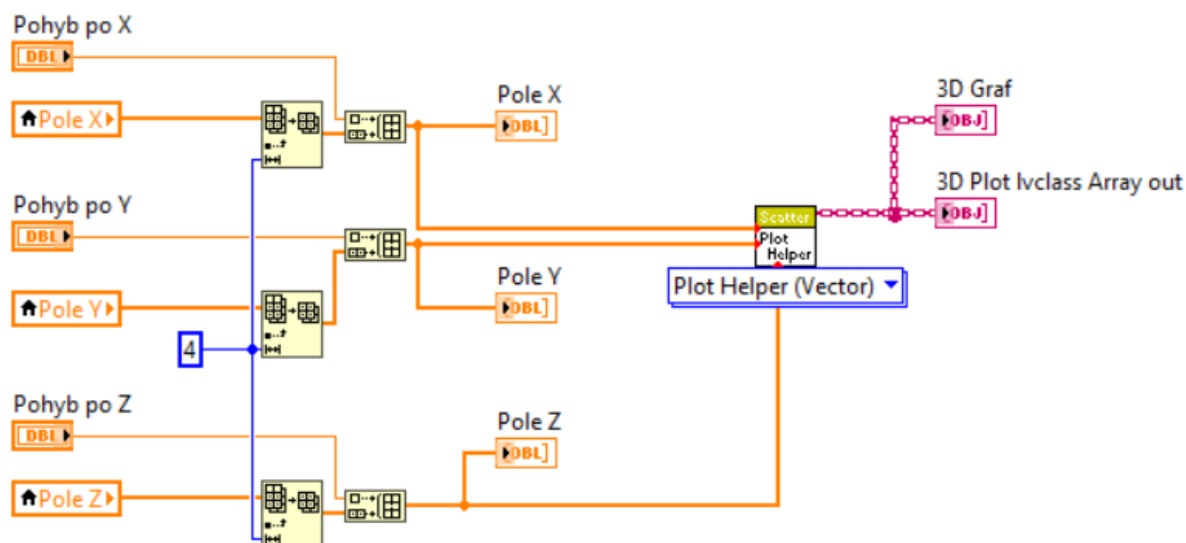
Vstupem do tohoto subVI jsou informace o obrazu, které jsou poskytovány z předešlého subVI, konkrétně „kostičkou“ *Particle Analysis*. Je upravovaná Area, kdy se ošetřuje, aby byl hledán opravdu největší objekt. Informace Bounding Rect obsahuje hraniční body nejvýraznějšího objektu, ty jsou následně použity při vytváření ohraničení vybraným tvarem a barvou na výsledném obraze.



Obrázek 39: SubVI pro vyznačení nejjasnějšího objektu

6.1.5 Vykreslení do grafu

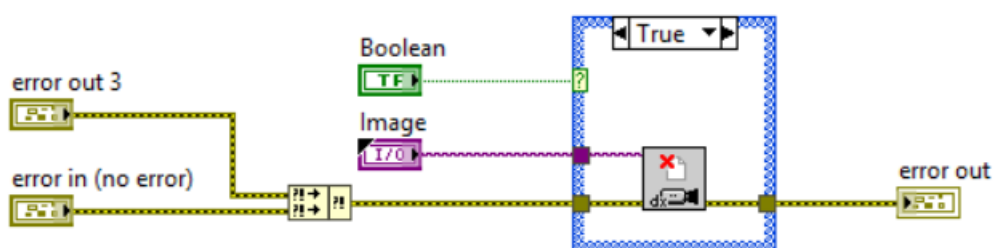
Toto subVI má za úkol vykreslovat do grafu posledních 5 hodnot (pozic). Vstupem jsou souřadnice aktuálních poloh, které jsou převáděny na pole, jelikož to vyžaduje „kostička“ pro vytvoření 3D grafu.



Obrázek 40: Vykreslení posledních 5 hodnot do 3D grafu

6.1.6 Nenačtení kamery a ukončení

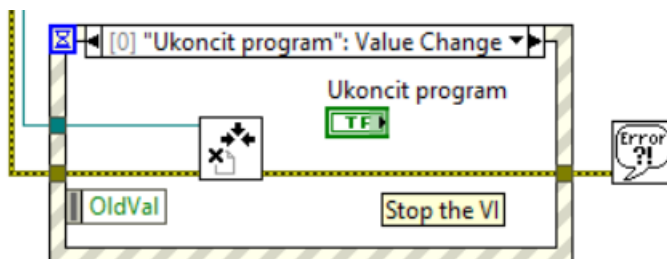
V případě, že kamera nebyla na začátku programu načtena, dochází k vykonání tohoto subVI a v chybovém klastru je vypsána příslušná chyba. Druhým případem, kdy dochází k vykonání tohoto subVI je pokud je zmáčknuto tlačítko *STOP*, viz. 6.1.7 Ukončení programu.



Obrázek 41: SubVI vypisující nenačtení kamery a konec programu

6.1.7 Ukončení programu

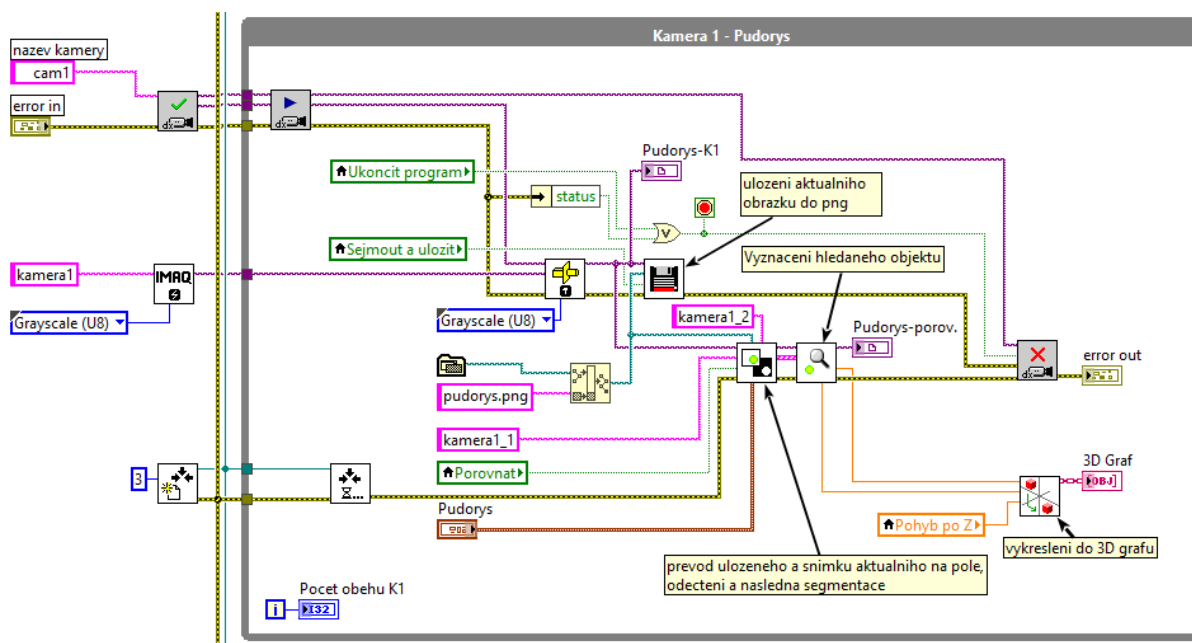
Ukončení programu je řízenou tzv. Event strukturou. Je to speciální struktura, která se používá pro nejrůznější reakce na určité události. Má konkrétní struktura reaguje na stisk tlačítka *UKONCIT PROGRAM*. Po stisku tohoto tlačítka je na selektor Case struktury přivedena hodnota TRUE a dochází k ukončení snímání.



Obrázek 42: Event struktura pro ukončení programu

6.2 Kód pro jednu kameru

Všechna výše zmíněná subVI jsou použita v hlavním VI takovým způsobem, aby vytvářela funkční aplikaci. Hodnoty, které je potřeba v programu předávat, jsou předávány pomocí lokálních proměnných, jak je vidět na obrázku níže. *Pudorys – K1* je displej, na němž se zobrazuje aktuální obraz v reálném čase. *Pudorys – porov.* je pak druhý displej, na kterém se zobrazuje obraz po průchodu segmentací a vyznačení nejvýraznějšího objektu. Názvy jednotlivých konstant, displejů a rozsahu pro prahování jsou specifické pro každou kameru. *Počet oběhu K1* je informativní hodnota, která počítá počet oběhu smyčky.



Obrázek 43: Kód pro jednu kameru

6.3 Funkčnost programu

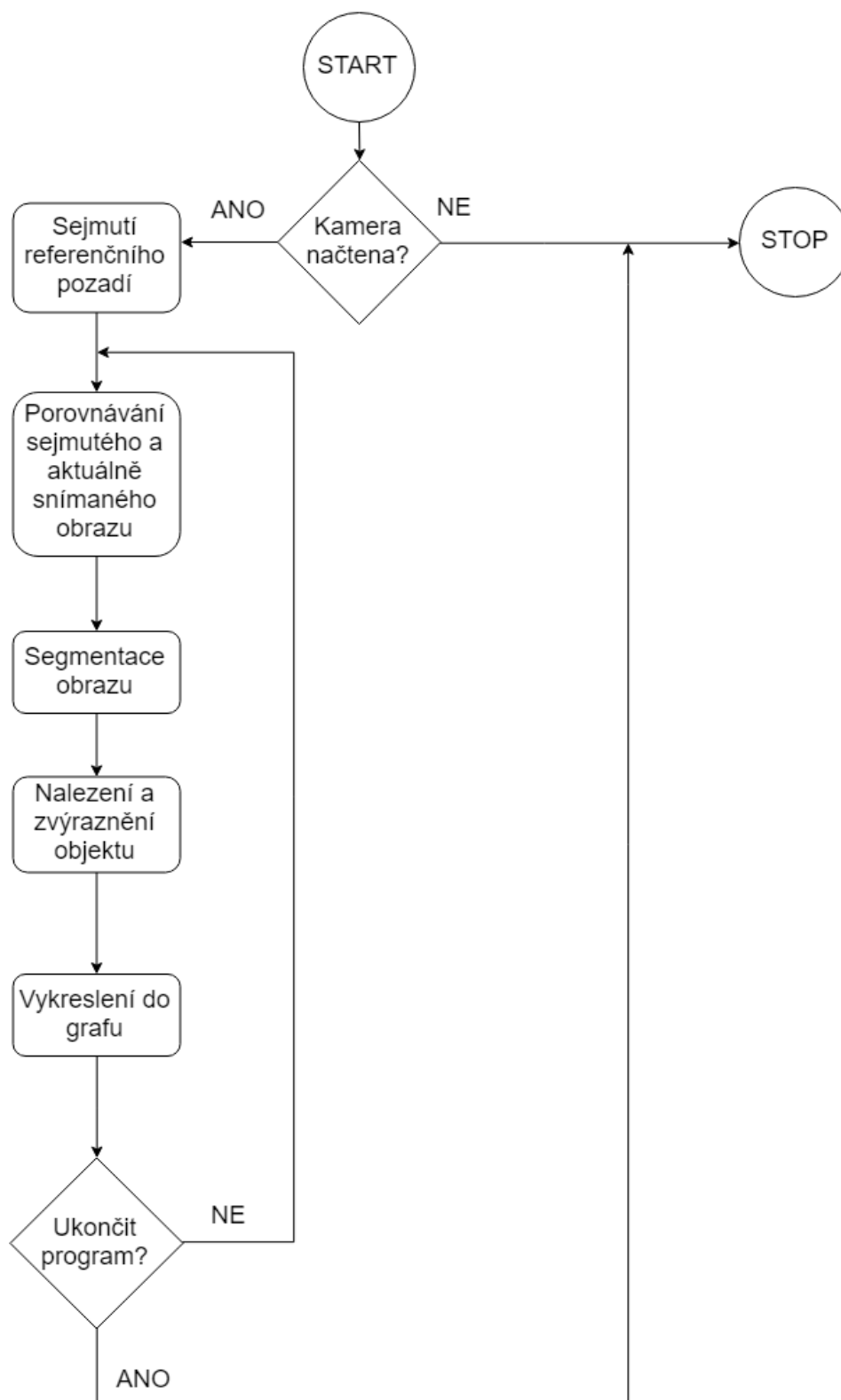
Aplikace, již jsem vytvořila, je složena ze 3 smyček while. Význam použití právě této smyčky je ten, že zajišťuje kontinuální běh programu, dokud není splněna určitá podmínka. V mém případě je to stisk tlačítka pro ukončení programu.

Jak jsem již zmínila výše při popisu jednotlivých subVI, po spuštění programu dochází nejprve k načtení a konfiguraci kamer. Následně je barevný obraz jdoucí z kamery převeden na obraz monochromatický, ve stupních šedi, a zobrazen na displej. Dále je nutno sejmut referenční pozadí, tedy aktuální obraz bez objektu, který má být detekován. Tento snímek je pomocí funkce *Application Directory* ukládán do složky s programem, díky tomu pak aplikace funguje bez problému komukoli na jakémkoliv počítači. Po uložení pozadí je do prostoru vkládán míček a je spuštěno subVI *Porovnání*. Jelikož každá kamera má trochu jiné světelné podmínky je rovněž i prahovací rozsah nastavován pro každou kameru zvlášť. Konkrétně tedy hodnota, od níž jsou pixely považovány za pixely hledaného objektu. Horní prahovací hodnota je vždy nastavena na 255. Po průchodu segmentací je obraz předáván do subVI pro vyznačení objektu. Odtud jsou vytahovány hodnoty souřadnic nejvýraznějšího objektu, tedy míčku a předávány do subVI pro vykreslení pohybu míčku do grafu. Rovněž je na displej zobrazován výsledný obraz s vyznačeným míčkem.

Důležitou částí tohoto programu je synchronizace. Jelikož je program složen ze 3 paralelních smyček dochází k tomu, že počet vykonání jednotlivých smyček není stejný a tedy snímky z kamery nejsou čteny najednou. Jednotlivé smyčky jsou mezi sebou synchronizovány pomocí synchronizačního mechanismu *Rendezvous*. To se dá přeložit jako setkání a tímto způsobem to také funguje. V kódu je definováno určité místo, v němž na sebe jednotlivé úlohy čekají, dokud se v tomto místě nesejde předem daný počet úloh, v tomto případě tedy 3. Základní funkčnost *Rendezvous* je tvořena třemi „kostičkami“. První je *Create Rendezvous.vi*, jež vytváří speciální proměnnou, která definuje, kolik úloh se musí ve vybraném místě sejít. Druhá „kostička“ je *Wait at Rendezvous.vi*, jež zajišťuje, aby na sebe dané úlohy počkaly. Poslední „kostička“ je *Destroy Rendezvous.vi*, která ruší vytvořenou proměnnou. Tato základní struktura se dá ještě rozšířit o další funkce, například pro zjištění, zda je dané *Rendezvous* ještě platné, nicméně tyto funkce jsem při své práci nevyužila.

Jak jsem zmínila v kapitole 4, maximální počet snímků za sekundu pro jednotlivé kamery je 30. Při běhu programu se ovšem tato hodnota pohybuje okolo 7 – 8 snímků. Přenos jednoho snímku pak trvá kolem 125 – 140 ms. S měnící se hodnotou snímku za sekundu se tento čas samozřejmě mění.

Funkčnost celého programu je zobrazena na Obrázek 44 ve formě vývojového diagramu. Hierarchie aplikace je poté zobrazena v příloze A. Je zde zobrazeno hlavní VI, v němž běží program, jednotlivá subVI, která jsou vytvořena a propojení mezi jednotlivými prvky a částmi.



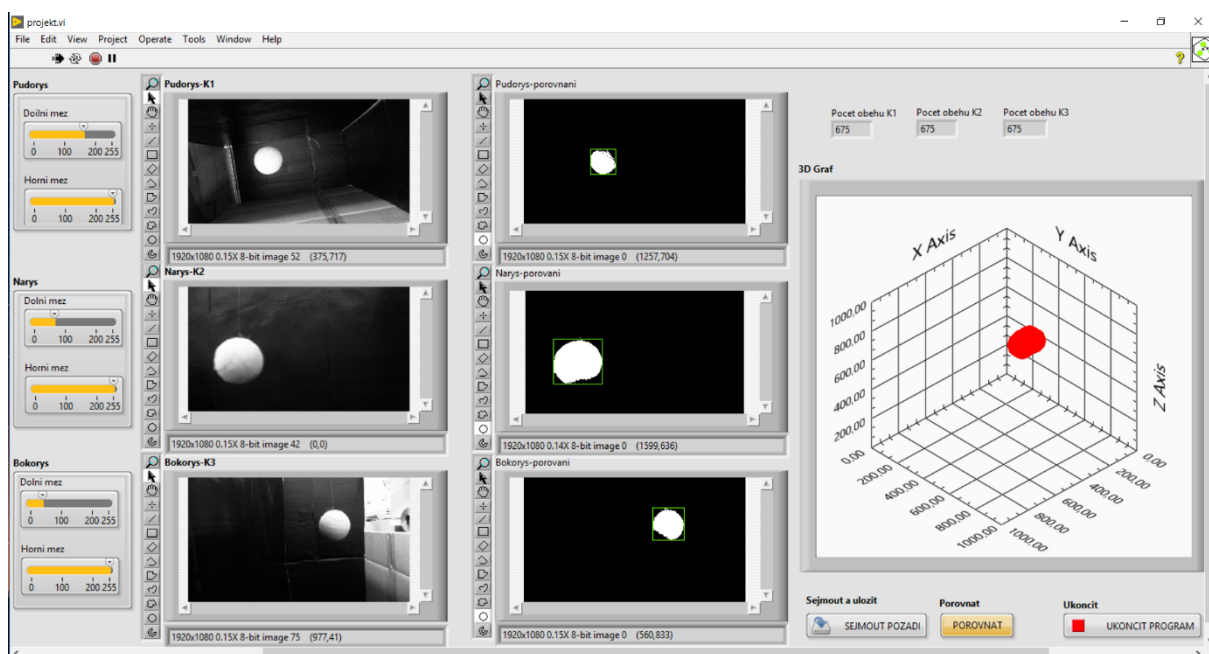
Obrázek 44: Vývojový diagram mé aplikace v LabVIEW

6.4 Čelní panel aplikace

Čelní panel mé aplikace, tedy uživatelské rozhraní je složen ze 6 displejů, kdy 3 z nich vykreslují aktuální obrazy jednotlivých kamer a vedle toho je analogicky tentýž aktuální obraz, nicméně po průchodu segmentací. Někdy se stává, že míček není na displeji vykreslen zcela dokonale. To je způsobeno tím, že aplikace vytěžuje procesor skoro na maximum, nehledě na fakt, že obraz z kamer je do počítače přenášen přes USB. V důsledku tohoto pak dochází ke zkreslení tvaru.

Další součástí mého GUI jsou ovládací prvky – slidery, které slouží k nastavování prahovacích hodnot. Dále jsou to jednotlivá tlačítka – *SEJMOUT A ULOZIT*, *POROVNAT* a *UKONCIT PROGRAM*. Nad těmito tlačítky se nachází 3D graf a nad ním pak počet oběhů jednotlivých smyček.

Ve skryté části čelního panelu jsou pak jednotlivé „displeje“ pro kontrolu případných chyb v kódu. Při výskytu chyby je vždy zobrazen její kód a stručný popis, co je v kódu špatně. Tyto chybové klastry slouží k odhalení chyb, které vznikají za běhu aplikace. Jelikož tento proces zpracování chyb respektuje data flow⁹, při zjištění chyby se již následující části kódu neprovádí, pouze je prostřednictvím chybového klastru předána do dalšího uzlu, dokud nedojde na konec řetězce, kde je zobrazena.



Obrázek 45: Uživatelské rozhraní

⁹Data flow - tedy datový tok znamená, že se vykonává ta část kódu, která má na svém vstupu platná data

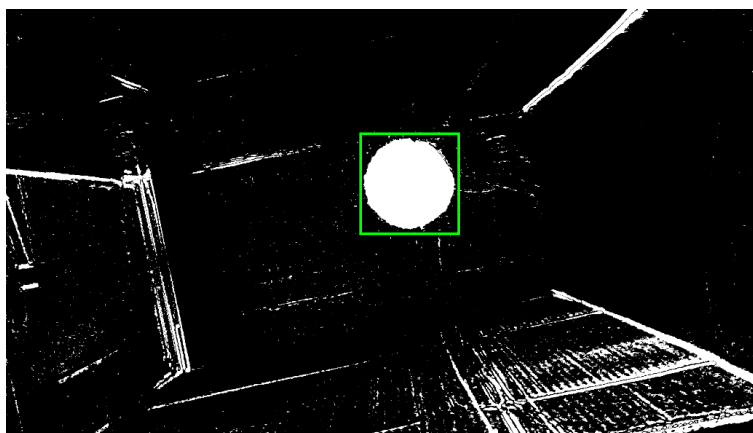
7 Práce s obrazem

Sedmá kapitola je věnována ukázce obrazu před a po prahování, kdy byla použita metoda prahování s více prahy, konkrétně s dvěma. Nicméně vyšší prahovací hodnota byla vždy nastavena na maximum, jelikož bylo předpokladem, že hledaný objekt je tvořen nejjasnějšími pixely na obraze. Dalo by se tedy říci, že byla použita metoda pouze s jedním prahem. Druhá část této kapitoly pak připadá výslednému vykreslení do grafu.

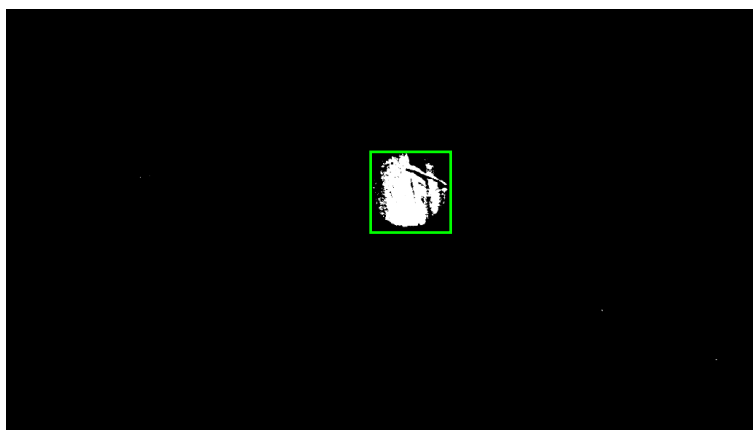
7.1 Výsledný obraz

Na Obrázek 45 výše je vidět, že pro každou kameru jsou na čelní panelu vždy 2 displeje. Pro ukázkou jsem vybrala několik situací, které jsem při své práci zaznamenala.

Nejprve bylo třeba správně nastavit prahy pro jednotlivé kamery. Při příliš nízké hodnotě se totiž stávalo, že na obrázku byl kromě hledaného objektu ještě zobrazován ještě nežádoucí šum – Obrázek 46. Naopak při vysokých prahovacích hodnotách nebyl hledaný objekt zcela zobrazen – Obrázek 47.

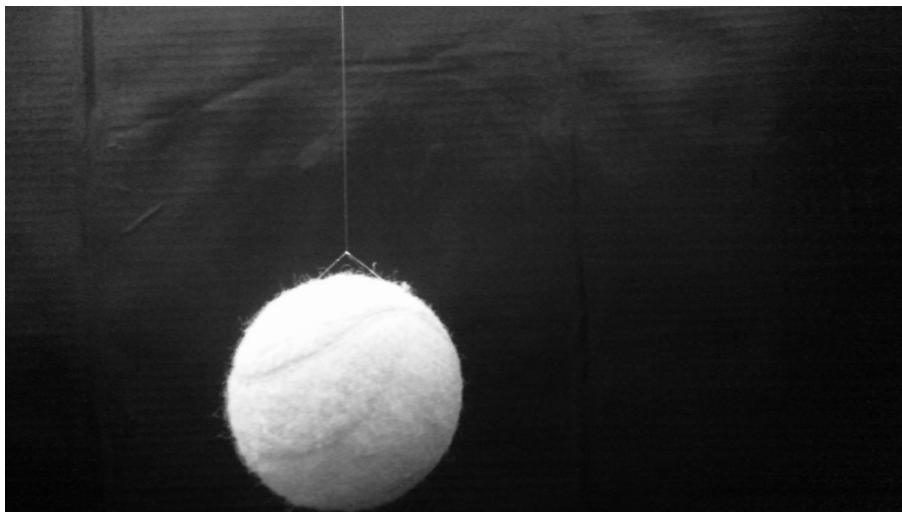


Obrázek 46: Snímek s nízkou prahovací hodnotou

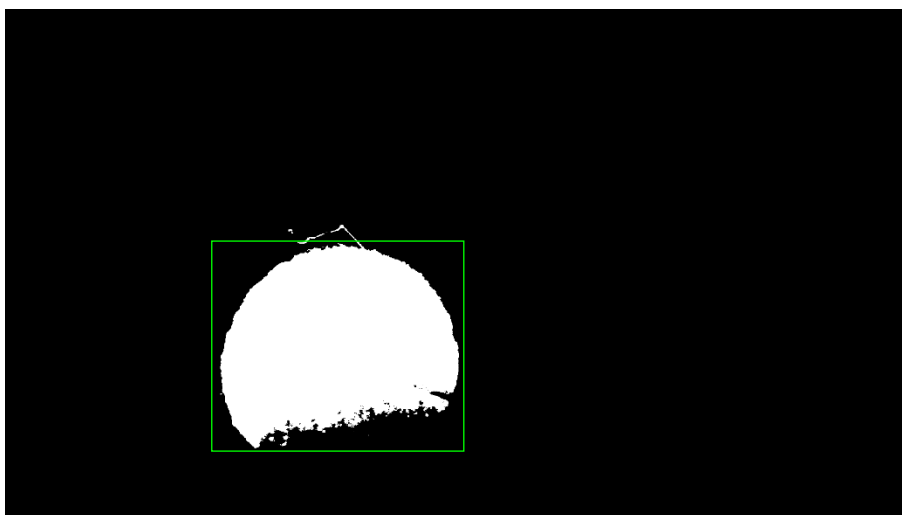


Obrázek 47: Snímek s vysokou prahovací hodnotou

Nastavení prahovacích hodnot při každém pokusu bylo rozdílné, jelikož okolní podmínky nebyly nikdy totožné a nikdy ideální. Na Obrázek 48 příklad z jedné kamery před průchodem prahováním a následně na Obrázek 49 tatáž scéna po průchodu prahováním.



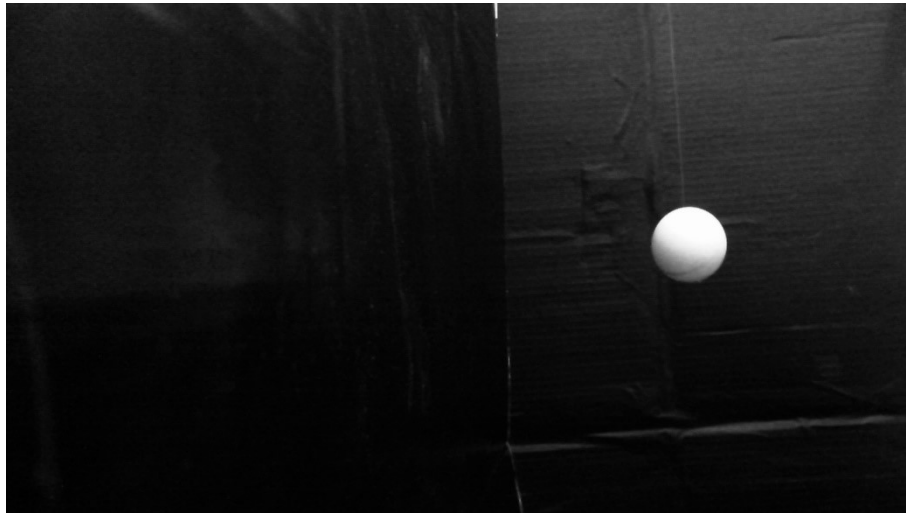
Obrázek 48: Snímek z čelní kamery před prahováním - nárys



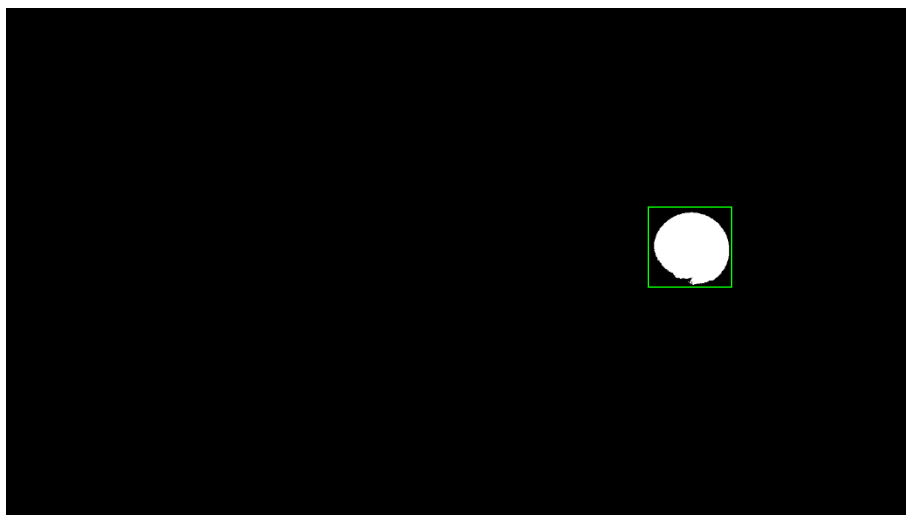
Obrázek 49: Snímek z čelní kamery po prahování - nárys

Na tomto příkladu lze dobře vidět, že spodní strana míčku je na původním snímku tmavší. To znamená, že pixely na tomto místě nabývají nižších hodnot a při prahování jsou určeny jako pozadí, nikoliv objekt.

Svou aplikaci jsem vyzkoušela i pro pingpongový míček bílé barvy o velikosti 40 mm. Jak jsem předpokládala, detekce a následné vykreslení proběhlo tak jak mělo. Ačkoliv je míček bílý a vůči černému pozadí dokonalé kontrastní, ani tvar tohoto míčku není vykreslen se 100% přesností. Příčina je stejná jako v případě tenisového míčku.



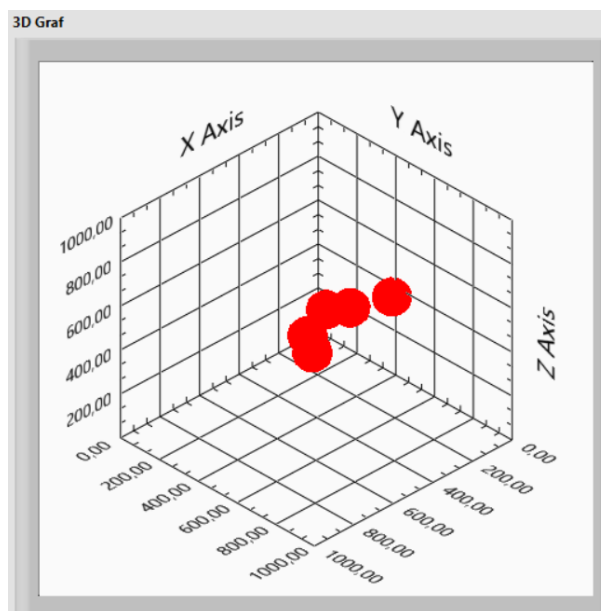
Obrázek 50: Snímek z boční kamery před prahováním - bokorys



Obrázek 51: Snímek z boční kamery po prahování - bokorys

7.2 Vykreslení do grafu

Z informací o obraze, jež jsou poskytovány, je vybírána tzv. *Center of Mass*. Do češtiny to lze přeložit jako střed hmoty, v tomto případě tedy střed detekovaného objektu. Tato informace je používána jako souřadnice pro vykreslení pohybu míčku do grafu. V grafu je vždy zobrazeno 5 posledních hodnot.



Obrázek 52: 3D graf

Tyto hodnoty jsou do grafu zapisovány v pixelech, tudíž jednotka použita v grafu jsou rovněž pixely. Pro praktické využití lze hodnoty z pixelů převést na milimetry. Je definováno, že 1 mm odpovídá přibližně 3,78 pixelům. Pokud by tedy konkrétní hodnota byla vydělena 3,78 byla by získána hodnota v milimetrech. Toto číslo by nicméně neodpovídalo úplně realitě, jelikož vždy dochází k určité odchylce.

První odchylkou, která je při měření určována je tzv. Absolutní odchylka měření Δ_X . Tato odchylka je určena ze vztahu:

$$\Delta_X = X_M - X_P, \quad (7.1)$$

kde X_M je reálná hodnota a hodnota X_P je skutečná hodnota. Absolutní odchylka měření je využívána pro určení, s jakou přesností je měření, v tomto případě tedy vykreslení, je provedeno.

Druhou určovanou odchylkou je Relativní odchylka měření δ_X . Výpočet je prováděn jako:

$$\delta_X = \frac{\Delta_X}{X_M} * 100. \quad (7.2)$$

Tato odchylka je určována v procentech a je používána pro určení přesnosti použité metody měření.

[43]

ZÁVĚR

Při vytváření této práce jsem se nejprve blíže teoreticky seznámila s kamerovými systémy a následně s metodami a technikami používanými při zpracování obrazu. Tyto nabyté znalosti jsem poté využila při vytváření své aplikace ve vývojovém prostředí LabVIEW.

Výsledkem mé práce je funkční program pro detekci a vykreslování objektu v reálném čase. Použití příslušného hardwaru bylo pro tuto práci dostačující, nicméně pokud by tato aplikace měla být prakticky využita, mělo by být použito kamer určených pro strojové vidění. Použitá konstrukce byla rovněž dostatečná a poskytla dost velký prostor ke snímání. Rozmístění kamer na konstrukci může být různé, ale měl by být dodržen pravý úhel mezi jednotlivými kamerami.

Kamery mají nastavenou snímkovací frekvenci 30 fps, reálně je ovšem tato hodnota proměnlivá a zpravidla není dosaženo ani 10 fps. Rychlost přenosu 1 snímku je vyjádřena jako 1/snímkovací frekvence. Kvalita, jaké jednotlivé snímky dosahují, je ovlivňována především osvětlením v místnosti, rychlostí, jakou je s míčkem v prostoru pohybováno a také jestli je při běhu aplikace na počítači spuštěno ještě něco jiného, tedy jak moc je vytěžován procesor. Od toho se rovněž odvíjí počet snímku za sekundu, se zvyšujícím se vytížením procesoru se počet snímků zmenšuje. Barevné rozlišení kamer je ovlivněno tím, že v kameře je použit CMOS snímač, který má při neideálních světelných podmínkách nižší citlivost a tedy horší barevné rozlišení. Při testování se toto horší barevné rozlišení potvrdilo, proto jsem se pro svou práci rozhodla využít konstantní pozadí černé barvy.

Jednotlivé snímky jsou ukládány do téže složky, kde se nachází vytvořený program. Snímky jsou černobílé a jsou ukládány v 8bitové hloubce ve formátu PNG. Hodnoty jednotlivých pixelů míčku jsou vždy v určitých místech nižší, míček tedy v těchto místech nabývá tmavších stupňů šedi a tyto části jsou poté určovány jako pozadí. To zapříčiňuje nepřesné vykreslení tvaru snímaného objektu. Tomuto faktu by se dalo předejít použitím podsvícení. Pro segmentaci obrazu je použita metoda prahování s více prahy, konkrétně dvěma. Hodnota druhého prahu je nastavována na maximum, jelikož hledaný objekt by měl mít nejvýraznější jasovou hodnotu. Hodnota prvního prahu značí, od jaké jasové hodnoty jsou již pixely určovány jako objekt. Defaultně je tato hodnota softwarem LabVIEW nastavována na 128.

Výsledný graf pro pohyb objektu má na všech osách maximální hodnotu 1000 px, z toho plyne, že je nastaven pouze na vykreslování míčku z určité oblasti. Dle výše zmíněného výpočtu lze určit, že tento prostor je velký přibližně 264,55 x 264,55 x 264,55 mm.

Tvorba této práce mi přinesla mnoho nových informací o práci s obrazy a především jsem se lépe naučila pracovat s vývojovým prostředím LabVIEW. Nejproblematictější částí pro mě bylo správné pochopení funkčnosti některých „kostiček“ a jejich následná implementace. Konečná verze mé aplikace je přiložena v IS EDISON spolu s videem spuštěného programu.

POUŽITÁ LITERATURA

- [1] MICHALEC, Libor. *30 let historie Labview* [online]. 2017 [cit. 2018-11-24]. Dostupné z: <https://vyvoj.hw.cz/30-let-historie-labview.html>
- [2] HALEŠ, Rostislav. *Postavte si měřicí systém s LabVIEW* [online]. 2014 [cit. 2018-11-25]. Dostupné z: <https://www.dps-az.cz/mereni/id:7945/postavte-si-merici-system-s-labview>
- [3] HOTAŘ, Vlastimil. *Úvod do problematiky strojového vidění*. V Liberci: Technická univerzita, 2015. ISBN 978-80-7494-156-6.
- [4] HOTAŘ, Vlastimil. *Úvod do problematiky strojového vidění*. V Liberci: Technická univerzita, 2015. ISBN 978-80-7494-202-0.
- [5] PEŠKA, Robert. *LabView 2018 zrychluje a posiluje integraci* [online]. 2018 [cit. 2018-11-26]. Dostupné z: <https://vyvoj.hw.cz/labview-2018-zrychluje.html>
- [6] PEŠKA, Robert. *LabView NXG na vlastní oči* [online]. 2017 [cit. 2018-11-26]. Dostupné z: <https://vyvoj.hw.cz/labview-nxg-na-vlastni-oci.html>
- [7] *LifeCam Studio* [online]. [cit. 2018-12-02]. Dostupné z: <https://www.microsoft.com/accessories/cs-cz/products/webcams/lifecam-studio/q2f-00004>
- [8] VÁCLAVÍK, Lukáš. *LifeCam Studio: našlapaná Full HD webkamera od Microsoftu* [online]. 2010 [cit. 2018-12-02]. Dostupné z: <https://www.cnews.cz/lifecam-studio-naslapan-a-full-hd-webkamera-od-microsoftu/>
- [9] ŠTĚPÁN, Filip. *Identifikace objektů v obraze* [online]. Brno, 2016 [cit. 2018-12-26]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=127466.
Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky.
- [10] *Cvičení 10 - Morfologické operace* [online]. [cit. 2018-12-26]. Dostupné z: http://midas.uamt.feec.vutbr.cz/ZVS/Exercise10/content_cz.php
- [11] PIHAN, Roman. *Gamut* [online]. [cit. 2018-12-29]. Dostupné z: http://www.fotoroman.cz/glossary/3_gamut.htm
- [12] *Seznámení s barevnými modely* [online]. [cit. 2018-12-29]. Dostupné z: http://product.corel.com/help/CorelDRAW/540240626/Main/CZ/Doc/wwhelp/wwhimpl/common/html/wwhelp.htm?context=CorelDRAW_Help&file=CorelDRAW-Understanding-color-models.html
- [13] *Barevné modely* [online]. [cit. 2018-12-29]. Dostupné z: <https://coptkm.cz/portal/reposit.php?action=0&id=40346&revision=-1&instance=1>

- [14] *Barevné modely* [online]. [cit. 2018-12-29]. Dostupné z: <http://www.ivt.mzf.cz/grafika/barevne-modely/>
- [15] *Color models* [online]. [cit. 2018-12-29]. Dostupné z: <https://software.intel.com/en-us/ipp-dev-reference-color-models>
- [16] *Color Models for MapInfo* [online]. 2002 [cit. 2018-12-29]. Dostupné z: https://georezo.net/jparis/MI_Enviro/Colors/color_models.htm
- [17] Barevný model. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-01-02]. Dostupné z: https://cs.wikipedia.org/wiki/Barevn%C3%BD_model
- [18] HORÁK, Karel. *Jasové transformace* [online]. [cit. 2018-12-30]. Dostupné z: http://midas.uamt.feec.vutbr.cz/ZVS/Lectures/05_Jasove_transformace.pdf
- [19] HLAVÁČ, Václav a Jan KYBIC. *Předzpracování v prostoru obrazu: Jasové transformace* [online]. [cit. 2018-12-30]. Dostupné z: <http://cmp.felk.cvut.cz/cmp/courses/33zsl1zima2005/slidy/jasoveTransformace.pdf>
- [20] *Cvičení 4 - Jasové transformace* [online]. [cit. 2019-01-02]. Dostupné z: http://midas.uamt.feec.vutbr.cz/ZVS/Exercise04/content_cz.php
- [21] PIHAN, Roman. *Histogram a jeho praktické využití* [online]. [cit. 2019-01-02]. Dostupné z: http://www.fotoroman.cz/tech1/exposure_histo.htm
- [22] *Homogenní souřadnice* [online]. [cit. 2019-01-02]. Dostupné z: https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=19976
- [23] HAVELKA, Jan. *Geometrické transformace obrazu* [online]. Brno, 2008 [cit. 2019-01-02]. Dostupné z: <https://dspace.vutbr.cz/xmlui/bitstream/handle/11012/55353/6290.pdf?sequence=2&isAllowed=y>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií, Ústav informačních systémů.
- [24] STRACHOTA, Pavel. *Geometrické transformace pomocí matic* [online]. 2. 4. 2010 [cit. 2019-01-05]. Dostupné z: http://saint-paul.fjfi.cvut.cz/base/sites/default/files/POGR/POGR2/07.maticove_transformace.pdf
- [25] POTŮČKOVÁ, Markéta. *Staré mapy TEMAP - eLearning* [online]. 2013 [cit. 2019-01-05]. Dostupné z: <http://web.natur.cuni.cz/gis/temap/images/elearning/Georeferencovn.pdf>
- [26] HLAVÁČ, Václav. *Hledání hran* [online]. Praha, 2005 [cit. 2019-04-09]. Dostupné z: <http://cmp.felk.cvut.cz/cmp/courses/33zsl1zima2005/slidy/DetekceHran.pdf>. Učební text. Fakulta elektrotechnická ČVUT v Praze katedra kybernetiky, Centrum strojového vnímání.

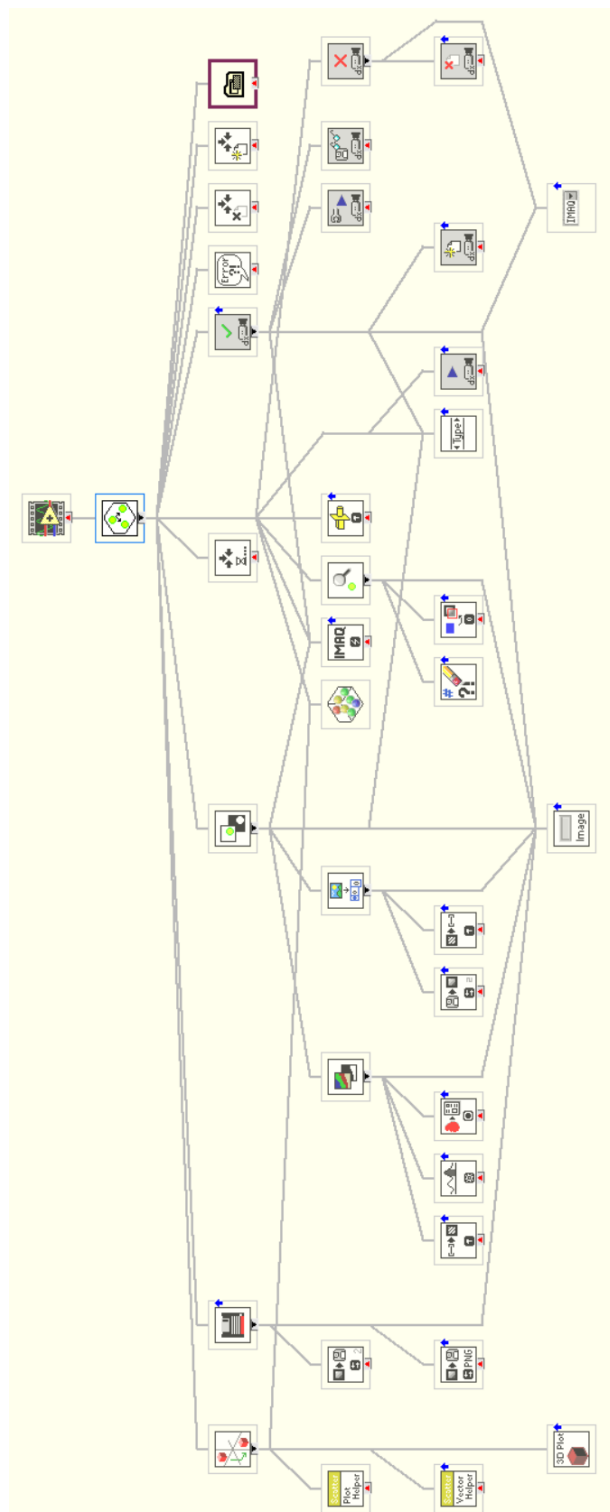
- [27] Klapal, Matěj. *Detekce rasistických symbolů z obrazů* [online]. Brno, 2015 [cit. 2019-04-09]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=105095.
Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací.
- [28] *Detekce hran, segmentace* [online]. [cit. 2019-04-09]. Dostupné z: https://kalabovi.org/pitel:msz:detekce_hran_segmentace
- [29] *Cvičení 8 - Detekce hran* [online]. [cit. 2019-03-21]. Dostupné z: http://midas.uamt.feec.vutbr.cz/ZVS/Exercise08/content_cz.php
- [30] Horák, Karel. *Matematická morfologie* [online]. [cit. 2019-03-21]. Dostupné z: http://midas.uamt.feec.vutbr.cz/ZVS/Lectures/11_Matematicka_morfologie.pdf
- [31] Krňoul, Zdeněk. *Zpracování digitalizovaného obrazu - Segmentace: Úvod, prahování* [online]. Plzeň [cit. 2019-03-03]. Dostupné z: <http://www.kky.zcu.cz/uploads/courses/zdo/prezentace/zdo-05.pdf>. Učební text. Západočeská univerzita, Fakulta aplikovaných věd, Katedra Kybernetiky.
- [32] Straka, Stanislav. *Segmentace obrazu* [online]. Brno, 2009 [cit. 2019-03-03]. Dostupné z: <https://is.muni.cz/th/tzp80/dp.pdf>. Diplomová práce. Masarykova univerzita, Fakulta informatiky.
- [33] Španěl, Michal a Vítězslav Beran. *Obrazové segmentační techniky: Přehled existujících metod* [online]. Brno, 2006 [cit. 2019-03-03]. Dostupné z: <http://www.fit.vutbr.cz/~spanel/segmentace/>. Studijní dokumentace. Vysoké učení technické v Brně, Fakulta informačních technologií, Ústav počítačové grafiky a multimediální.
- [34] Janáková, Ilona. *Segmentace* [online]. Brno [cit. 2019-03-03]. Dostupné z: http://midas.uamt.feec.vutbr.cz/POV/Lectures/05_Segmentace.pdf. Učební text. Vysoké učení technické v Brně.
- [35] Železný, Miloš. *Zpracování digitalizovaného obrazu* [online]. Plzeň [cit. 2019-04-09]. Dostupné z: http://www.kky.zcu.cz/uploads/courses/zdo/ZDO_aktual_130215.pdf. Učební text. Západočeská univerzita, Fakulta aplikovaných věd, Katedra Kybernetiky.
- [36] Hlaváč, Václav. *Šedotónová matematická morfologie* [online]. Praha [cit. 2019-04-11]. Dostupné z: <http://people.ciirc.cvut.cz/~hlavac/TeachPresCz/11DigZprObr/71-06MatMorfolGrayCz.pdf>. Učební text. České vysoké učení technické v Praze, Centrum strojového vnímání.
- [37] Keilová, Lucie. *Úvod do matematické morfologie* [online]. Brno, 2016 [cit. 2019-04-11]. Dostupné z: <https://is.muni.cz/th/tdfg8/BP.pdf>. Bakalářská práce. Masarykova univerzita, Přírodovědecká fakulta, Ústav matematiky a statistiky. Vedoucí práce prof. RNDr. Jiří Rosický, DrSc.

- [38] KOLOUCHOVÁ, Michaela. *Morfologické operace ve zpracování obrazu* [online]. Brno, 2008 [cit. 2019-04-11]. Dostupné z: <https://core.ac.uk/download/pdf/44387378.pdf>. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií, Ústav počítačové grafiky a multimédií. Vedoucí práce Ing. Adam Herout, Ph.D.
- [39] Kamerové systémy. *Laddin* [online]. 25. 4. 2014 [cit. 2019-04-12]. Dostupné z: http://www.ladinn.cz/ostatni/technika/kamerovy_system.html
- [40] IP-vs-analog-kamery-a-základní-pojmy. *Stasanet* [online]. [cit. 2019-04-12]. Dostupné z: <https://www.stasanet.cz/IP-vs-analog-kamery-a-zakladni-pojmy/>
- [41] KASAL, Jaroslav. Cesta do hlubin digitální kamery - jak pracuje digitální kamera. *PCWorld* [online]. 1. 2. 2002 [cit. 2019-04-12]. Dostupné z: <https://pcworld.cz/hardware/cesta-do-hlubin-digitalni-kamery-jak-pracuje-digitalni-kamera-12943>
- [42] MLČOCH, Vladimír. *Bezpečnostní kamerový systém CCTV* [online]. Brno, 2012 [cit. 2019-04-12]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=53342. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce Ing. Patrik Babnič.
- [43] SLÁMA, Adam. *Měření polohy objektu v prostoru pomocí dvojice kamer* [online]. Brno, 2017 [cit. 2019-04-29]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=152543. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce Ing. Petr Kříž.
- [44] *Microsoft LifeCam Studio: Technical Data Sheet* [online]. Microsoft Corporation, 2014 [cit. 2019-04-29]. Dostupné z: https://media.flixcar.com/f360cdn/Microsoft-303383578-Microsoft-TDS_LifeCamStudio.pdf

Seznam příloh

A Hierarchie aplikace v LabVIEW	66
B DAR0025_Aplikace_a_video	67

A Hierarchie aplikace v LabVIEW



B DAR0025_Aplikace_a_video

Tato příloha je uložena v IS EDISON a obsahuje mou aplikaci včetně použitých subVI. V této složce jsou také uloženy snímky používány jako pozadí a video, na němž je zaznamenána funkční spuštěná aplikace.